mtpiiexamenesresueltos.pdf

aGT1999

Análisis de Algoritmos

2º Grado en Ingeniería Informática

Escuela Politécnica Superior
Universidad Autónoma de Madrid

KEEP
CALM
AND
ESTUDIA
UN POQUITO

Metod. y Tecnol. de la Programación II, Febrero 1996

Apellidos:

Nombre:

1	2	3	4	5	6	7	T

1. Construir razonadamente el árbol de decisión del algoritmo de inserción.

2. Construir razonadamente el árbol AVL para la lista 13 9 12 1 4 6 8 16.
   Dar los árboles que resultan de extraer del AVL anterior los elementos 9, 12, 6 en ese orden.

3. Deducir por inducción que si T es un 2–árbol con N nodos internos, T tiene entonces $N+1$ hojas.

4. De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i-ésimo es $P(K == T[i]) = \frac{1}{C_N} i^2 \log i$, donde C_N es una constante de normalización. Calcular razonadamente el coste medio de las búsquedas con éxito que sobre la misma efectúe el algoritmo de búsqueda lineal.

5. Para un cierto método de hash H mediante direccionamiento abierto se puede demostrar que $A_H^f(N,m) = \frac{1}{1-\lambda} + \log(\frac{1}{1-\lambda}) + o(1)$. ¿Cuál será entonces el valor aproximado de $A_H^e(N,m)$?

6. (a) Dar razonadamente las tablas next y de desplazamientos del patrón binario 1 0 1 1 0 1 1 1 para el método KMP de izquierda a derecha.

   (b) De un patrón binario de 8 bits y que comienza y termina por 0 se sabe que su tabla next para el método KMP de izquierda a derecha es 0 0 0 1 1 1 2 3. Identificar razonadamente dicho patrón.

   (c) Se desea buscar el patrón de la cuestión (a) sobre una cierta cadena mediante el algoritmo de Boyer–Moore. Decidir razonadamente cuál sería un tamaño razonable de alfabeto y cuál sería el desplazamiento básico de cada carácter del mismo.

7. El siguiente pseudocódigo recursivo recoge esencialmente el algoritmo tradicional de cálculo del determinante de una matriz A por los adjuntos de la primera columna (se supone que `Adj(i,j)` es una cierta función que construye el menor adjunto al elemento `a[i][j]`):

```
float Det(matriz A, dim N)
  si N == 1:
    devolver a[1][1];
  else:
    det = 0;
    sig = 1;
    para i de 1 a N:
       b   = Det (Adj(i,1), N-1);
      det += sig * a[i][1] * b;
      sig  = -sign;
   devolver det;
```

(a) Encontrar razonadamente la ecuación en recurrencias que gobierna el número de productos que dicho algoritmo realiza.

(b) Resolver razonadamente dicha ecuación.

$P(1)=0$; $P(N)= 2+N\,P(N-1)$?

ó

$P(N)= N\,(P(N-1)+2)$?

Metod. y Tecnol. de la Programación II, Noviembre 1997

Apellidos: **Nombre:**

1	2	3	T

1. Se quiere aplicar el algoritmo habitual de búsqueda lineal sobre una tabla T de N elementos sabiendo que la probabilidad de que haya que buscar la clave $T[i]$, $1 \leq i \leq N$ es $P(K = T[i]) = (i^2 \log i)/C_N$, donde C_N es una constante que asegura que $\sum_1^N P(K = T[i]) = 1$.

   Estimar razonadamente y con la mayor precisión posible el crecimiento de $A_{BLin}(N)$.

2. Estimar el crecimiento de la función $T(N)$ de argumento entero que verifica la desigualdad recurrente $T(N) \leq N + 2T(\lfloor N/2 \rfloor)$, donde $T(1) = 0$.

3. El siguiente pseudocódigo es una ligera variante del habitualmente utilizado en la ordenación por Inserción:

```
InsertSort(tabla T, dim N)
  para i de 2 a N:
    j=i;
    mientras j > 0 y T[j-1] > T[j]:
      intercambiar T[j-1] y T[j];
      j--;
  volver;
```

Utilizando el intercambio de elementos como operación básica, calcular razonadamente $W_{InserSort}(N)$ y $A_{InserSort}(N)$.

Metod. y Tecnol. de la Programación II, Diciembre 1997

Apellidos: **Nombre:**
Grupo:

1	2	3	T
			10

1. El árbol de decisión para el algoritmo de ordenación por inserción dado más abajo siempre comienza por la comparación `1:2`. Encontrar razonadamente la mitad derecha de su árbol de decisión para $N = 4$, indicando tras cada comparación el estado de la tabla.

```
InsertMax(tabla T, dim N)
  para i de 2 a N:
    j = i;
    mientras j > 1 y T[j-1] > T[j]:
      intercambiar T[j-1] y T[j];
      j--;
```

2. A. Dar la evolución de la rutina `Partir` utilizada en el algoritmo QuickSort sobre la tabla `[5, 8, 1, 9, 3, 11, 4]`.

   B. Crear el AVL asociado a la lista `[1, 3, 4, 5, 8, 9, 10]` y eliminar de la misma los elementos `5, 3, 8` (tras cada una de estas eliminaciones se ha de obtener un árbol binario de búsqueda, pero no necesariamente un AVL).

3. El pseudocódigo inferior implementa un algoritmo de búsqueda sobre tablas generales (no necesariamente ordenadas). Indicar brevemente el modo de funcionamiento del algoritmo y dar razonadamente su árbol de decisión para $N = 4$.

```
indice BLinRaro(tabla T, ind P, ind U, clave K)
  i = P;
  j = U;
  mientras i <= j:
    si T[i] == K:
      devolver i;
    else si T[j] == K:
      devolver j;
    else:
      i++;
      j--;
  devolver error;
```

Metod. y Tecnol. de la Programación II, Febrero 1997

Apellidos: **Nombre:**

1	2	3	4	5	6	7	8	T

1. ¿Cuántas comparaciones de clave hará como mínimo un algoritmo de ordenación local sobre la permutación de $N = 3K$ elementos (2K+1, K+1, 1, 2K+2, K+2, 2, ..., 3K, 2K, K)?

2. Resolver razonadamente la recurrencia $T(N) = N + 9T(N/3)$, $T(1) = 0$.

3. De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es $P(K == T[i]) = \frac{1}{C_N}\frac{\log i}{i^2}$, donde C_N es una constante de normalización. Calcular razonadamente y con la mayor precisión posible el coste medio de las búsquedas con éxito que sobre la misma efectúe el algoritmo de búsqueda lineal.

4. Construir razonadamente el árbol de decisión para tablas de 3 elementos de la siguiente versión del algoritmo de ordenación de la burbuja, indicando tras cada comparación la posición relativa de los elementos de la tabla y el valor de las variables `camb` y `lim`:

```
BbjSort(tabla T, dim N)
  camb = 1;
  lim  = 1;
  mientras lim < N y camb == 1:
    camb = 0;
    i = N;
    mientras i > lim:
      si T[i] > T[i-1]:
        intercambiar T[i] y T[i-1];
        camb = 1;
      i--;
    lim++;
```

Dar también el árbol de decisión si en el código se eliminan las sentencias con el indicador de intercambios `camb`.

5. a. Construir razonadamente un AVL para la lista 7 6 8 1 3 4 10 9.
   b. Construir razonadamente el heap asociado a la lista 9 10 4 3 1 8 6 7.

6. (a) Dar un patrón binario de ocho bits que comience y acabe por 0 y cuya tabla next para el algoritmo KMP de izquierda a derecha sea 0 0 0 1 2 0 1 1.

   (b) Se desea encontrar el patrón `mima` en la cadena `mi mama me mima` mediante el método de Boyer-Moore simple. Calcular los desplazamientos básicos de los distintos caracteres del alfabeto y dar razonadamente la evolución del método al aplicarse a las cadenas anteriores.

7. El siguiente pseudocódigo corresponde a una rutina recursiva de cálculo de los números de Fibonacci:

```
long FibRec(int N)
  si N==0 o N==1:
    devolver 1;
  else:
    devolver FibRec(N-1) + FibRec(N-2);
```

Determinar razonadamente cuántas sumas efectuará dicho método para calcular el N–ésimo número de Fibonacci

11072 THEORY OF AUTOMATA AND FORMAL LANGUAGES I

The main goal of this course is to give an introduction to the three fundamental types of mechanisms for the definition of languages: regular expressions, free context grammars, and general grammars, as well as the corresponding types of machines for the recognition of languages: finite automata, push-down automata and Turing machines. The fundamental properties of the families of languages defined by them are also studied, together with the necessary conditions for a language being of one of the previous types.

The course can be used as an introduction to the study of compilers of computer languages, which is the subject of the course Language Processors, that corresponds to the third course in Computer Engineering.

(Second year; Second semester. Mandatory: Final exam)
Professors: Roberto Moriyón, Pilar Rodríguez.

11073 STATISTICS

Probability. Random variables. Random vectors. Most common probability models. Random sampling. Confidence Interval. Testing of statistical hypothesis. X2 tests.

(Second year; Second semester. Mandatory: Final exam)
Professors: Mayte Rodríguez, A. Cuevas.

11074 BUSINESS ECONOMICS

General Introduction to business: Administration, Accounting, Investment, Finances, Production, Supplies, Sales, Marketing, Human Resources.

(Second year; Second semester. Optional: Final exam)
Professors: Fernando Maestre, Cristina Esteban.

11330 COMPUTER ARCHITECTURE

This course is designed for students with a basic knowledge in computer systems. The aim of this course is concerned with the design techniques used to improve the performance of computer systems. The evolution of computer architectures from their former design to nowadays are introduced using real examples to illustrate the material.
This course is complemented with the courses Computer Technology and Structure Iand II.

(Third year, first semester, Mandatory: Final Exam)
Professors: Javier Garrido, Francisco Gómez, Francisco Gómez, Julio Faura

11331 ARTIFICIAL INTELLIGENCE

The aim of this course is to give students an introduction on artificial intelligence. The first part of the course introduces the LISP programming language. A second part introduces the concept of production systems, so as the basic searching strategies and algorithms (A*, Y/O*). Finally, propositional and first order predicate logic are introduced, as well as rule based systems and automatic plan generation.
This course is complemented with course 11337, Knowledge Engineering.

(Third year; First semester. Mandatory: Final exam)
Professor: Francisco Saiz, Javier Contreras

11332 COMPUTER NETWORKS I

The contents of this matter are related with the three first layer of the OSI model : Physical, Data Link and Network layer. Transmission media like cooper, fibre optics and air are compared. Conventional and digital telephone networks are studied in the Physical layer. Several protocols related with the frame and the error recovery, are the subjects of the data link layer. The differences between several LAN , Ethernet,

Metodología y Tecnología de la Programación, Febrero 1999

Apellidos: **Nombre:**

1	2	3	4	5	6	7	T

1 Se sabe que $T_1 \approx f$ y que $T_2 = o(f)$, con T_1, T_2 y F crecientes. Decidir razonadamente la verdad o falsedad de las siguientes afirmaciones:
(a) $T_1 + T_2 \approx f$ (b) $T_1 T_2 \approx f$ (c) $T_2/T_1 = \Theta(1)$ (d) $T_1/T_2 = \Omega(1)$ $\lim \frac{T_1}{f} = 1$ $\lim \frac{T_2}{f} = 0$

2. El algoritmo de ordenación de la burbuja efectúa en primer lugar la comparación de T[1] con T[2], y suponiendo que T[1] < T[2], compara entonces T[2] con T[3]. Dar para $N = 4$ el subárbol del árbol de decisión del algoritmo obtenido suponiendo que T[1] < T[2] y que T[2] > T[3]. Utilizar el siguiente pseudocódigo inferior e indicar en cada paso el estado de la tabla y su parte ordenada.

```
bubSort(tabla T, dim N)
  n_elem = N;
  mientras n_elem > 1 :
    para i de 1 a n_elem-1:
      si T[i] > T[i+1]:
        intercambiar T[i] y T[i+1];
    n_elem--;
```

3 Se aplica el método de búsqueda lineal sobre una tabla T para la que se verifica que $P(K == T[i]) = \frac{1}{C_N}\frac{\log i}{i}$, donde C_N es una constante de normalización. Estimar razonadamente el número medio de comparaciones de clave en las búsquedas con éxito.

4 a) Dar razonadamente la evolución de la inserción sucesiva en una tabla hash de direccionamiento abierto de tamaño 11 de los elementos 25, 12, 16, 11, 33, 23, 34 usando tanto sondeos lineales como cuadráticos. Situar los elementos dados en una tabla horizontal cuyos índices vayan de 0 a 10).
b) Los elementos anteriores se desean insertar en una tabla hash con encadenamiento. Decidir razonadamente cuál podría ser un tamaño mínimo de tabla que garantice que el número de colisiones en búsquedas con y sin éxito sea menor que 3.

5 Justificar precisa y razonadamente el coste medio de algoritmo de inserción tal y como se recoge en el pseudocódigo inferior cuando se toma como operación básica el intercambio de claves y se aplica sobre tablas de N elementos.

```
InsertSort(tabla T, dim N)
  para i de 2 a N:
    j=i;
    mientras j > 1 y T[j-1] > T[j]:
      intercambiar T[j-1] y T[j];
      j--;
  volver;
```

6 El algoritmo inferior proporciona para entradas M y K la parte entera de $\log_K(M)$. Dar razonadamente y con la mayor precisión posible una estimación superior como función de M del número de sumas realizado en el cálculo de logEntero(M, K) (escribir primero la desigualdad recurrente adecuada).

```
int logEntero(float M, float K)
  si M < 1 :      devolver error ;
  else si M < K : devolver 0 ;
  else :          devolver 1 + logEntero(M/K, K) ;
```

7. a. Encontrar razonadamente para el patrón binario P = [1 0 1 1 0 1 1] sus tablas next y de desplazamiento. Dar razonadamente la evolución del algoritmo KMP de izquierda a derecha cuando en el mismo se intenta hallar el patron anterior en la cadena S = [1 0 0 1 0 1 0 1 0 1 1 1 0 1 1 0 1 1], indicando las comparaciones erróneas y sus desplazamientos subsiguientes.
b. Calcular las tablas de desplazamientos básicos del método de Boyer–Moore para el patrón mimi y dar razonadamente la evolución de dicho método para encontrar el patrón anterior en la cadena mi mama mima a mimi.

HECHO en los ejercicios

Metod. y Tecnol. de la Programación II, Noviembre 1999

Apellidos:

Nombre: Grupo:

1	2	3	4	T

1. a. Definir qué se entiende por la notaciones $f = o(g)$, $f = O(g)$.
   b. Suponiendo fijado para un algoritmo A un espacio de entradas $E_A(N)$ de tamaño N, dar con precisión la definicición del coste en los casos peor y medio de A.
   c. Dar los valores de $A_{IS}(N)$ y $W_{BB}(N)$ para los algoritmos de inserción IS y Busqueda binaria BB.
   d. Dar la definición de algoritmo de ordenación local.

2. De unas funciones positivas y crecientes T_1, T_2 y f se sabe que $T_1 \approx f$ y $T_2 = o(f)$. Establecer razonadamente la verdad o falsedad de las siguientes afirmaciones:
   a. $T_1 + T_2 \approx f$
   b. $T_1 T_2 \approx f$
   c. $T_2/T_1 = o(1)$

3. Los N elementos de una tabla T verifican que $P(K == T[i]) = \sqrt{i}/C_N$, donde C_N es una constante de normalización. Determinar razonadamente y con la mayor precisión posible el valor del número medio $A^e_{BLin}(N)$ de comparaciones de clave que efectúa sobre la tabla T el algoritmo de Busqueda Lineal para encontrar una cualquiera de las claves $T[i]$.

4. Determinar razonadamente el número mínimo de comparaciones de clave que sobre la tabla de $N = 3K$ elementos

$$(2K+1, \ldots, 3K, 2K, \ldots, K+1, K, \ldots, 1)$$

realizarían los métodos de Inserción o de la Burbuja.

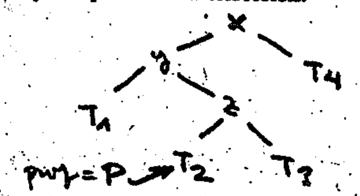
HECHO EN LOS EJERCICIOS

Metod. y Tecnol. de la Programación II, Diciembre 1999

Apellidos: **Nombre:** **Grupo:**

1	2	3	4	T

1. a. Dar la definición de árbol AVL.
   b. Enumerar las primitivas del TAD Diccionario.
   c. Indicar mediante la notación Θ los rendimientos en los casos medio y peor de los algoritmos MergeSort, QuickSort y HeapSort.
   d. A partir del esquema inferior, indicar las transformaciones necesarias para corregir el desequilibrio indicado, razonando por qué se produce esa corrección.



2. El siguiente pseudocódigo nos da una implementación del método de la burbuja.

```
BbjSort(tabla T, dim N)
  i = N ;
  swap = 1 ;
  mientras swap == 1 e i > 1 :
    swap = 0 ;
    para j de 1 a i -1 :
      si T[j] > T[j+1] :
        intercambiar T[j] y T[j+1] ;
        swap = 1 ;
    i-- ;
```

Dar el subárbol de su árbol de decisión para $N = 4$ que recoge las cdc realizadas por el algoritmo sobre aquellas tablas donde el primer elemento es menor que el segundo y éste es mayor que el tercero.

3. Dar razonadamente una estimación del crecimiento de la función T tal que $T(1) = 0$ y que cumple la siguiente desigualdad recurrente

$$T(N) \leq N\sqrt{N} + 2T(\lfloor N/4 \rfloor).$$

4. a. Dar la evolución de la rutina `Partir` utilizada en el algoritmo QuickSort sobre la tabla `[4, 7, 1, 9, 2, 10, 3]`.

   b. Crear el AVL asociado a la lista `[1, 2, 4, 6, 7, 9, 12]` y eliminar de la misma los elementos `4, 2, 7` (tras cada una de estas eliminaciones se ha de obtener un árbol binario de búsqueda, pero no necesariamente un AVL.

Nº 2019

15 pts.

Metod. y Tecnol. de la Programación II, Noviembre 2000

Apellidos: **Nombre:**
Grupo:

1	2	3	T

NOTA IMPORTANTE: sólo se tendrán en cuenta aquellas respuestas debidamente razonadas

1. a. Definir con precisión qué se entiende con la expresión $f = g + O(h)$.

   b. Enunciar los rendimientos en los casos mejor, medio y peor de los algoritmos de Inserción y Selección.

   c. Definir qué se entiende por permutación traspuesta.

   d. Si $\mathcal{A}$ es un cierto algoritmo que recibe entradas I cuyo tamaño se representa por $|I|$, definir qué se entiende por espacio de entradas de $\mathcal{A}$ y por su caso medio.

2. i. Se sabe que $T_1 \approx f$, $T_2 \approx g$ y $g = o(f)$. Decidir razonadamente la verdad o falsedad de las siguientes afirmaciones (dar un breve argumento en las ciertas y un contraejemplo o contraargumento en las falsas):

   (a) $T_1 + T_2 \approx f$;

   (b) $\sqrt{T_1} + T_2 \approx \sqrt{f}$.

   ii. Se aplica un algoritmo local A de ordenación a la permutación de $N = 2K$ elementos
   `(2K K 2K-1 K-1 2K-2 K-2 ... K+2 2 K+1 1)`.

   Expresar en función de N cuántas comparaciones de clave efectuará A sobre dicha permutación.

3. De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es

$$P(K = T[i]) = \frac{1}{C_N}\frac{1}{\sqrt{i}},$$

   donde C_N es una constante que garantiza que $\sum_1^N P(K = T[i]) = 1$.

   Calcular razonadamente y con la mayor precisión posible el coste medio de las búsquedas lineales realizadas con éxito.

1

#2.

a. (i) Se tiene $\frac{T_1+T_2}{f} = \frac{T_1}{f} + \frac{T_2}{g}\frac{g}{f} \to 1$, con $\frac{T_1}{f}\to 1$, $\frac{T_2}{g}\to 1$, $\frac{g}{f}\to 0$

y entonces $T_1+T_2 \approx f$.

(ii) es falso: si $T_1 = f = N^2$, $T_2 = g = N^{3/2}$,

se tiene que $g = o(f)$, pero $\sqrt{f} + g = N + N^{3/2} \approx N^{3/2}$

y, por tanto, es falso.

b. calculamos las inversiones de la permutación:

$\sigma = (2k \;\; k \;\; 2k-1 \;\; k-1 \;\; \dots \;\; k+2 \;\; 2 \;\; k+1 \;\; 1)$

inv: $2k-1 \;\; k-1 \;\; 2k-3 \;\; k-2 \;\; \dots \;\; 3 \;\; 1 \;\; 1$

① $= 1+3+5+7+\dots+2k-3+2k-1$

$= 1+(2+1)+(2\cdot 2+1)+(2\cdot 3+1)+\dots+2(k-2)+1$

$+2(k-1)+1$ (k unos)

$= k + 2\sum_{1}^{k-1} i = k+(k-1)k = k^2$

② $= 1+\dots+k-1 = \frac{k^2}{2} - \frac{k}{2}$

por tanto $n_a(\sigma) \geq inv(\sigma) = \frac{3k^2}{2} - \frac{k}{2} = \frac{3}{8}N^2 - \frac{N}{4}$

#3.

Se tiene $A_{BL}(N) = \frac{S_N}{C_N}$, con

① $S_N = \sum_1^N i\frac{1}{\sqrt{i}} = \sum_1^N i^{1/2}$

② $C_N = \sum_1^N \frac{1}{\sqrt{i}} = \sum_1^N i^{-1/2}$

Estimamos ambas sumas por separado

① como $\sqrt{x}\uparrow$, $\int_0^N x^{1/2}dx \leq S_N \leq \int_1^{N+1} x^{1/2}dx$

$\Rightarrow \left[\frac{2}{3}x^{3/2}\right]_0^N = \frac{2}{3}N^{3/2} \leq S_N \leq \frac{2}{3}(N+1)^{3/2} - \frac{2}{3}$

Metod. y Tecnol. de la Programación II, Diciembre 2000

Apellidos: **Nombre:**
Grupo:

1	2	3	T

NOTA IMPORTANTE: sólo se tendrán en cuenta aquellas respuestas debidamente razonadas

1. a. Enunciar la ecuación recurrente del número de comparaciones de clave efectuado por QuickSort.
   b. Enunciar las ventajas del algoritmo HeapSort frente otros algoritmos de ordenación por comparación de claves.
   c. Enunciar cuáles son y qué hacen las primitivas del TAD Diccionario.
   d. ¿Cuál es el número mínimo de nodos de un AVL de profundidad P?
2. Estimar el crecimiento de un función $T(N)$ que verifica la siguiente desigualdad recurrente:

$$T(N) \leq N^2 + T(\lfloor \frac{N}{2} \rfloor);$$
$$T(1) = 0.$$

3. Dar razonadamente el árbol de decisión para tablas de 3 elementos del siguiente algoritmo de ordenación:

```
RaroSort(tabla T, ind P, ind U)
  si T[U-1] < T[U] :
    MergeSort(T, P, U) ;
  else :
    HeapSort(T, P, U) ;
```

2 : 3
MS HS

User: jasiguen
Host: minerva
Class: minerva
Job: 5113668.spl

Nº 2096 50pts.

Metodología y Tecnología de la Programación II, Febrero 2001

Apellidos: **Nombre:**
Grupo:

1	2	3	4	5	6	7	T

NOTA IMPORTANTE: NO SE TENDRÁN EN CUENTA RESPUESTAS NO RAZONADAS

1. Para una tabla T con N elementos, se sabe que $P(K = i) = 1/(C_N i^{1/3})$, con C_N una constante de normalización. Calcular razonadamente el número medio de comparaciones de clave que se harán en búsqueda lineal para encontrar con éxito K en $T[]$.

2. Estimar ajustada y razonadamente el crecimiento de la función T que satisface $T(1) = 0$, así como la desigualdad recurrente $T(N) \leq N + 2T(\lfloor N/4 \rfloor)$.

3. a. ¿Cuántas hojas tiene un árbol de decisión sobre tablas de N elementos? Si $\mathcal{A}$ es un algoritmo de ordenación por comparación de claves, ¿qué relación hay entre su árbol de decisión para tablas de N elementos y $W_{\mathcal{A}}(N)$ y $A_{\mathcal{A}}(N)$?

   b. El método de Selección compara en primer lugar el elemento $T[1]$ con $T[2]$ y si $T[1]$ es menor, lo compara a continuación con $T[3]$. Suponiendo que $T[1] < T[2]$ y $T[1] > T[3]$, dar razonadamente la parte restante del árbol de decisión de dicho método para permutaciones de 4 elementos.

4. a. Definir qué se entiende por un Max–heap. Dar la evolución del algoritmo Heap Sort sobre la lista 1, 2, 3, 4, 5, 6, recogiendo la misma sobre árboles binarios.

   b. ¿Cuál es el redimiento del algoritmo RadixSort sobre una tabla de N elementos formados cada uno por K caracteres tomados de un alfabeto de M caracteres? Dar la evolución de RadixSort sobre la lista inferior, obtenida del alfabeto $\Sigma = \{1, 2, 3\}$, indicando en cada paso qué elementos se usan para ello:
   (1 2 3 1), (2 3 1 2), (3 1 2 3), (2, 1, 1, 3), (3, 1, 2, 1), (1, 1, 3, 2).

5. a. Dar la relación existente para un método H de hashing con direccionamiento abierto entre el coste medio de búsquedas fallidas $A^f_H(N,m)$ y el de búsquedas con éxito $A^e_H(N,m)$.

   b. Un método H de hashing con direccionamiento abierto cumple que $A^f_H(N,m) = 1/(1-\lambda)$. ¿Cuánto vale entonces $A^e_H(N,m)$? Si para dicho método y una tabla de 100 datos se quiere que el número medio de sondeos, tanto en búsquedas con éxito como sin éxito, no sea mayor que 2, ¿cuál es un tamaño de tabla adecuado?

6. a. Indicar qué función cumplen en el método KMP las tablas `next[]` y `d[]`. ¿Cuál es el rendimiento del método KMP en el caso peor cuando se usa para buscar un patrón de M caracteres en una cadena de N caracteres.

   b. Dar razonadamente las tablas `d[]` y `n[]` utilizadas por el algoritmo KMP para el patrón `P = [1 0 1 0 1 1 1]` primero sin tener y luego teniendo en cuenta el carácter de `P` que causa el error.

   c. Dar la evolución de KMP si se aplica para buscar el patrón P anterior en la cadena `S = [1 0 1 1 0 0 1 0 1 0 1 1 1]`. ¿Cuántas comparaciones de caracteres se hacen?

7. El siguiente algoritmo proporciona la posición del elemento mínimo de una tabla T entre las posiciones P y U:

```
pos MinRec(tabla T, pos P, pos U)
   si P == U: devolver P ;
   M = (P+U)/2 ;            // division entera
   m1 = MinRec(T, P, M) ;
   m2 = MinRec(T, M+1, U) ;
   si T[m1] < T[m2] : devolver m1 ;
   else :             devolver m2 ;
```

Estimar razonadamente el trabajo del algoritmo sobre una tabla de $U - P + 1 = N$ elementos. Para ello, definir una operación básica adecuada, establecer una ecuación para dicho trabajo, resolverla en algún caso particular adecuado y, finalmente, dar la solución general.

1

1- $A^e_{BL}(N) = \sum_{i=1}^{N} n_{BL}(T[i]) \cdot P(k=T[i])$

$\underbrace{\phantom{n_{BL}(T[i])}}_{i}$ ← justificándolo (si no, -1 pto)

$= \sum_{1}^{N} i \frac{1}{i^{1/3}} \cdot \frac{1}{C_N} = \frac{1}{C_N} \sum_{1}^{N} i^{2/3}$

$\frac{S_N}{C_N}$

$C_N \equiv$ cte de normalización, cumple:

$\sum_{1}^{N} \frac{1}{C_N i^{1/3}} = 1 \Rightarrow C_N = \underbrace{\sum_{1}^{N} \frac{1}{i^{1/3}}}$

f. decreciente, acotamos la suma con integrales

$\int_0^N \frac{1}{i^{1/3}} di \geq \sum_{1}^{N} \frac{1}{i^{1/3}} \geq \int_1^{N+1} \frac{1}{i^{1/3}} di$

$\frac{3}{2} i^{2/3} \Big]_0^N \geq \sum_{1}^{N} \frac{1}{i^{1/3}} \geq \frac{3}{2} i^{2/3} \Big]_1^{N+1}$

$\sum_{1}^{N} \frac{1}{i^{1/3}} \simeq \frac{3}{2} N^{2/3}$ ← justificarlo (si no, -1 pto)

La otra suma: creciente

$\int_0^N i^{2/3} di \leq \sum_{1}^{N} i^{2/3} \leq \int_1^{N+1} i^{2/3} di$

$\frac{3}{5} i^{5/3} \Big]_0^N \leq \sum_{1}^{N} i^{2/3} \leq \frac{3}{5} i^{5/3} \Big]_1^{N+1}$

$\sum_{1}^{N} i^{2/3} \simeq \frac{3}{5} N^{5/3}$ ← justificarlo (si no, -1 pto)

Por tanto $A_{BL} \simeq \frac{3/5\, N^{5/3}}{3/2\, N^{2/3}} = \frac{2}{5} N$

justificarlo

②

a) Analizamos para el caso particular $N=4^k$

$$T(N) \leq N + 2T(N/4) \leq N + 2\left(\frac{N}{4} + 2T\left(\frac{N}{4^2}\right)\right) = N + \frac{N}{2} + 2^2 T\left(\frac{N}{4^3}\right)$$

$$\leq N + \frac{N}{2} + \frac{N}{4} + 2^3 T\left(\frac{N}{4^3}\right) \dots$$

$$T(N) \leq \sum_{i=0}^{k-1} \frac{N}{2^i} + 2^k \cancel{T\left(\frac{N}{4^k}\right)} \to 0$$

$$= N \sum_{i=0}^{k-1} \frac{1}{2^i} = N \left(\frac{\frac{1}{2}^k - 1}{\frac{1}{2} - 1}\right) = 2N - 2N\left(\frac{1}{2^k}\right)$$

Hasta aquí 3 puntos

$$N = 4^k \to \log_2 N = 2k \to k = \frac{\log_2 N}{2}$$

$$T(N) \leq 2N - 2N\left(\frac{1}{\sqrt{N}}\right) = 2N - 2\sqrt{N}$$

Hasta aquí 5 puntos

fon creciente

b) $N=1 \to T(1) = 0$

Suponemos cierto para todo valor menor que N

$$T(N) \leq N + 2T\left(\left\lfloor\frac{N}{4}\right\rfloor\right) \leq N + 2\left(2\frac{N}{4} - 2\sqrt{\frac{N}{4}}\right) =$$

$$= N + \frac{4N}{4} - \frac{4}{2}\sqrt{N}$$

$$\boxed{T(N) \leq 2N - 2\sqrt{N}}$$

Hasta aquí 10 puntos.

$$\Rightarrow \quad 1 \leq \frac{S_N}{\frac{2}{3} N^{3/2}} \leq \underbrace{\left(\frac{N+1}{N}\right)^{3/2}}_{\downarrow\, 1} - \underbrace{\frac{1}{N^{3/2}}}_{\downarrow\, 0},$$

$$y \quad S_N \approx \frac{2}{3} N^{3/2}$$

② como $x^{-1/2} \downarrow$ $\quad \int_0^N x^{-1/2}dx \geq C_N \geq \int_1^{N+1} x^{-1/2}dx$

$$\Rightarrow \left[2\sqrt{x}\right]_0^N = 2\sqrt{N} \geq C_N \geq 2\sqrt{N+1} - 2$$

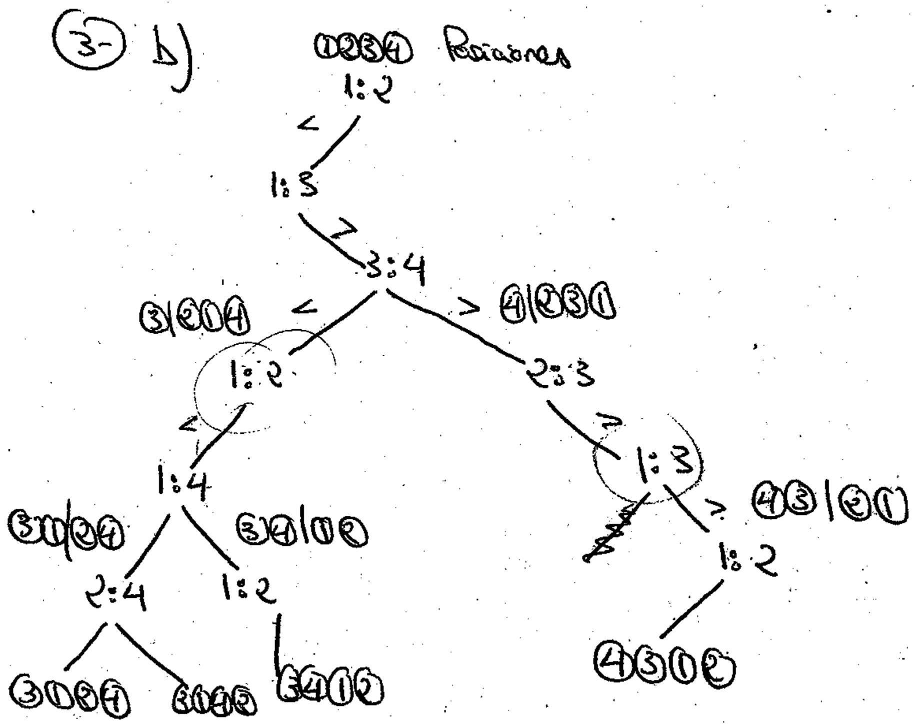
$$\Rightarrow \quad 1 \geq \frac{C_N}{2\sqrt{N}} \geq \underbrace{\frac{\sqrt{N+1}}{\sqrt{N}}}_{\downarrow\, 1} - \underbrace{\frac{1}{\sqrt{N}}}_{\downarrow\, 0} \Rightarrow C_N \approx 2\sqrt{N}$$

Esto sugiere que $A_{BL}(N) \approx \dfrac{\frac{2}{3} N\sqrt{N}}{2\sqrt{N}} = \dfrac{1}{3} N$

lo que comprobamos como sigue:

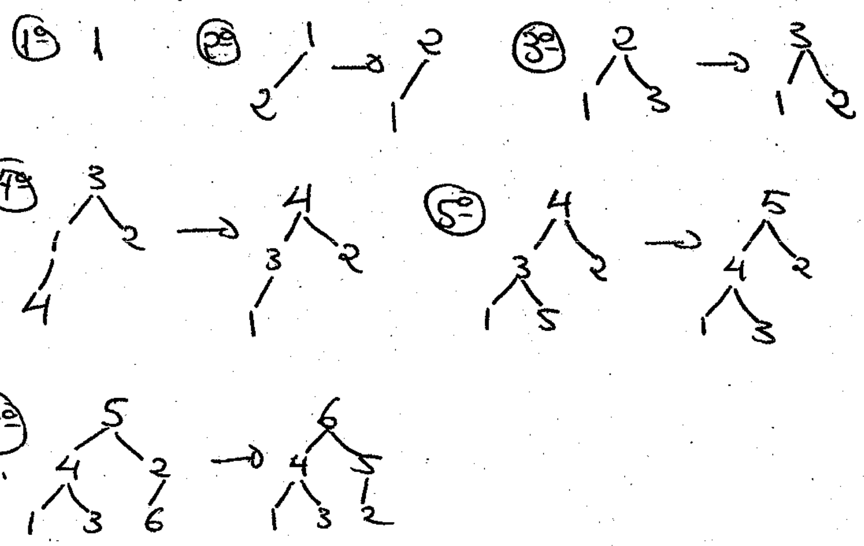
$$\frac{A_{BL}(N)}{N/3} = \underbrace{\frac{S_N}{\frac{2}{3} N^{3/2}}}_{\to 1} \cdot \frac{1}{\underbrace{C_N/2\sqrt{N}}_{\to 1}} \longrightarrow 1.$$

3- b)



Cada nodo incorrecto -1 pto.

4- a) Crear Heap

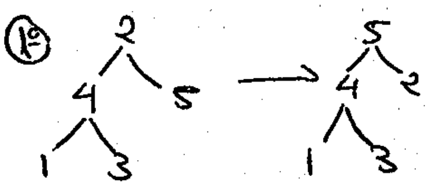


Cada paso incorrecto -1 pto

Atrás →

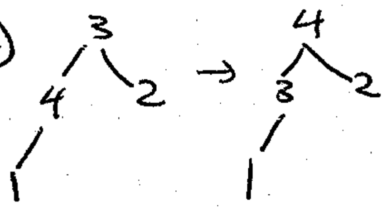
b) Ahora Heap

1º



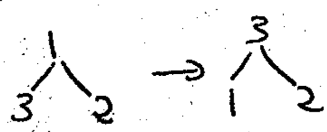
6

2º



5 6

3º



4 5 6

4º

3 4 5 6

5º

1

2 3 4 5 6

6º

1 2 3 4 5 6

#4.b El alfabeto tiene 3 caracteres, y los datos a ordenar tienen 4 ⑤

$\Rightarrow$ hacen falta 3 colas auxiliares (además de la principal) y se tienen 4 pasos

Paso (1) Q_1: (1 2 3 1) (3 1 2 1)
Q_2: (2 3 1 2) (1 1 3 2)
Q_3: (3 1 2 3) (2 1 1 3)
se unen en orden $Q_1\ Q_2\ Q_3$

Paso (2) Q_1: (2 3 1 2) (2 1 1 3)
Q_2: (3 1 2 1) (3 1 2 3)
Q_3: (1 2 3 1) (1 1 3 2)
—"—

Paso (3) Q_1: (2 1 1 3) (3 1 2 1) (3 1 2 3) (1 1 3 2
Q_2: (1 2 3 1)
Q_3: (2 3 1 2)

Paso (4) Q_1: (1 1 3 2) (1 2 3 1)
Q_2: (2 1 1 3) (2 3 1 2)
Q_3: (3 1 2 1) (3 1 2 3)

que unidas en orden $Q_1\ Q_2\ Q_3$ dan la ordenación buscada

Teoría: 1 pto

7

(Teo: 2 p)

#5.b. La relación entre A_H^e y A_H^f es

(4 p)

$$A_H^e(n,m) \simeq \frac{1}{n}\sum_0^{n-1} A_H^f(i,m)$$

En el problema

$$A_H^e(n,m) \simeq \frac{1}{n}\sum_0^{n-1}\frac{1}{1-i/m} \simeq \frac{1}{n}\int_0^n \frac{dx}{1-\frac{x}{m}}$$

$$= \frac{m}{n}\int_0^{n/m}\frac{du}{1-u} = \frac{1}{\lambda}\int_0^{\lambda}\frac{du}{1-u}$$

$$= \frac{1}{\lambda}\left[-\log(1-u)\right]_0^{\lambda} = \frac{1}{\lambda}\left(-\log(1-\lambda)\right)$$

$$= \frac{1}{\lambda}\log\frac{1}{1-\lambda}$$

5.c Se tiene $n=100$ y se quiere

(4 p) (i) $A_H^f(100,m) = \dfrac{1}{1-\frac{100}{m}} \leq 2$

(ii) $A_H^e(100,m) = \dfrac{1}{\frac{100}{m}}\log\dfrac{1}{1-\frac{100}{m}} \leq 2$

Resolver (i) es sencillo, pues se tiene

$\frac{1}{2} \leq 1-\frac{100}{m} \Rightarrow \frac{100}{m} \leq \frac{1}{2} \Rightarrow m \geq 200$ (o $\lambda \leq \frac{1}{2}$)

y como $A_H^e(n,m)$ es $\uparrow$ como función de λ, si $\lambda \leq 1/2$,

$$A_H^e(100,m) \leq \frac{1}{1/2}\log\frac{1}{1-1/2} = 2\log 2 \leq 2$$

Luego $m \geq 200$ es suficiente para

(i) (ii)

#6. $P = [1\ 0\ 1\ 0\ 1\ 1\ 1]$ | Teoría 2 ptos |

- $j=2$: $\begin{matrix} 1 & 0 & * \\ & 1 & 0 & 1 \end{matrix}$ } $n=0,\ d=2$
- $j=3$: $\begin{matrix} 1 & 0 & 1 & * \\ & 1 & 0 & 1 & 0 \end{matrix}$ } $n=1,\ d=2$

7

- $j=4$: $\begin{matrix} 1 & 0 & 1 & 0 & * \\ & & 1 & 0 & 1 & 0 & 1 \end{matrix}$ } $n=2,\ d=2$
- $j=5$: $\begin{matrix} 1 & 0 & 1 & 0 & 1 & * \\ & & 1 & 0 & 1 & 0 & 1 & 1 \end{matrix}$ } $n=3,\ d=2$
- $j=6$: $\begin{matrix} 1 & 0 & 1 & 0 & 1 & 1 & * \\ & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{matrix}$ } $n=1,\ d=5$

□ Teniendo en cuenta el carácter que causa el error: si $p[j] = p[n[j]]$,

hacemos $n'[j] = n'[n[j]]$

$d'[j] = d[j] + d'[n[j]]$

- Posiciones afectadas:

$j=2$: $n'[2] = n'[0] = 0$, $d'[2] = d[2] + d'[0] = 3$

$j=3$: $n'[3] = n'[1] = 0$, $d'[3] = d[3] + d'[1] = 2+1 = 3$

$j=4$: $n'[4] = n'[2] = 0$, $d'[4] = d[4] + d'[2] = 2+3 = 5$

- Tablas finales:

j	n	d	n'	d'
0	0	1	0	1
1	0	1	0	1
2	0	2	0	2
3	1	2	0	3
4	2	2	0	5
5	3	2	3	2
6	1	5	1	5

3 ptos (n, d) — 2 puntos (n', d')

3 puntos

- Test usando n', d'.

S 1 0 1 1 0 0 1 0 1 0 1 1 1

P 1 0 1 0_3 ⇒ 4 cdc, $d = 3$

1 0 1_2 ⇒ 8 cdc, $d = 3$

n' 1 1 0 1 0 1 1 1 ⇒ 9 cdc

n

8

Ent wando n, d

1 0 1 0 1 0 1 0

1 0 1 0_3... ⇒ 4 cdc, $d=2$

1 0_1... ⇒ 1 cdc, $d=1$

1 0 1_2... ⇒ 3 cdc, $d=2$

1_0 0... ⇒ 1 cdc, $d=1$ ⇒ 1 cdc

1 0 1 0 1 1 1 ⇒ 7 cdc

total: 16 cdc

#7. minRec divide una tabla de N elementos en 2, una con $\lfloor N/2 \rfloor$ y otra con $\lceil N/2 \rceil$ elementos. Hace 1 cdc en $T[m1] < T[m2]$

OB: 1 pto
Ec Rec: 3 pto

∴ si $n(N)$ = nº cdcs de minRec en tabla de N elementos, $n(N) = 1 + n(\lfloor N/2 \rfloor) + n(\lceil N/2 \rceil)$

$n(1) = 0$

(3p) a. caso particular: $N = 2^k$, entonces

$$n(N) = 1 + 2n(N/2) = 1 + 2 + 2^2 n(N/2^2)$$

$$= 1 + 2 + \dots + 2^{k-1} + 2^k \underbrace{n(N/2^k)}_{=0} = \sum_0^{k-1} 2^j$$

$$= 2^k - 1 = N - 1$$

(3p) b. En general: inducción

$n(N) = N-1$ si $N=1$, y si $n(K) = K-1$ si $K < N$

$$n(N) = 1 + n(\underbrace{\lfloor N/2 \rfloor}_{\hat{N}}) + n(\underbrace{\lceil N/2 \rceil}_{\hat{N}}) = 1 + \lfloor N/2 \rfloor - 1 +$$

$$\lceil N/2 \rceil - 1$$

$$= \lfloor N/2 \rfloor + \lceil N/2 \rceil - 1 = N - 1$$

Metodología y Tecnología de la Programación II, Febrero 2001

Apellidos: **Nombre:**
Grupo:

1	2	3	4	5	6	7	T

NOTA IMPORTANTE: NO SE TENDRÁN EN CUENTA RESPUESTAS NO RAZONADAS

1. Para una tabla T con N elementos, se sabe que $P(K = i) = 1/(C_N i^{1/3})$, con C_N una constante de normalización. Calcular razonadamente el número medio de comparaciones de clave que se harán en búsqueda lineal para encontrar con éxito K en $T[]$.

2. Estimar ajustada y razonadamente el crecimiento de la función T que satisface $T(1) = 0$, así como la desigualdad recurrente $T(N) \leq N + 2T(\lfloor N/4 \rfloor)$.

3. a. ¿Cuántas hojas tiene un árbol de decisión sobre tablas de N elementos? Si $\mathcal{A}$ es un algoritmo de ordenación por comparación de claves, ¿qué relación hay entre su árbol de decisión para tablas de N elementos y $W_{\mathcal{A}}(N)$ y $A_{\mathcal{A}}(N)$?

   b. El método de Selección compara en primer lugar el elemento $T[1]$ con $T[2]$ y si $T[1]$ es menor, lo compara a continuación con $T[3]$. Suponiendo que $T[1] < T[2]$ y $T[1] > T[3]$, dar razonadamente la parte restante del árbol de decisión de dicho método para permutaciones de 4 elementos.

4. a. Definir qué se entiende por un Max–heap. Dar la evolución del algoritmo Heap Sort sobre la lista 1, 2, 3, 4, 5, 6, recogiendo la misma sobre árboles binarios.

   b. ¿Cuál es el redimiento del algoritmo RadixSort sobre una tabla de N elementos formados cada uno por K caracteres tomados de un alfabeto de M caracteres? Dar la evolución de RadixSort sobre la lista inferior, obtenida del alfabeto $\Sigma = \{1, 2, 3\}$, indicando en cada paso qué elementos se usan para ello:

   (1 2 3 1), (2 3 1 2), (3 1 2 3), (2, 1, 1, 3), (3, 1, 2, 1), (1, 1, 3, 2).

5. a. Dar la relación existente para un método H de hashing con direccionamiento abierto entre el coste medio de búsquedas fallidas $A_H^f(N,m)$ y el de búsquedas con éxito $A_H^e(N,m)$.

   b. Un método H de hashing con direccionamiento abierto cumple que $A_H^f(N,m) = 1/(1-\lambda)$. ¿Cuánto vale entonces $A_H^e(N,m)$? Si para dicho método y una tabla de 100 datos se quiere que el número medio de sondeos, tanto en búsquedas con éxito como sin éxito, no sea mayor que 2, ¿cuál es un tamaño de tabla adecuado?

6. a. Indicar qué función cumplen en el método KMP las tablas `next[]` y `d[]`. ¿Cuál es el rendimiento del método KMP en el caso peor cuando se usa para buscar un patrón de M caracteres en una cadena de N caracteres.

   b. Dar razonadamente las tablas `d[]` y `n[]` utilizadas por el algoritmo KMP para el patrón `P = [1 0 1 0 1 1 1]` primero sin tener y luego teniendo en cuenta el carácter de P que causa el error.

   c. Dar la evolución de KMP si se aplica para buscar el patrón P anterior en la cadena `S = [1 0 1 1 0 0 1 0 1 0 1 1 1]`. ¿Cuántas comparaciones de caracteres se hacen?

7. El siguiente algoritmo proporciona la posición del elemento mínimo de una tabla T entre las posiciones P y U:

```
pos MinRec(tabla T, pos P, pos U)
   si P == U: devolver P ;
   M = (P+U)/2 ;              // division entera
   m1 = MinRec(T, P, M) ;
   m2 = MinRec(T, M+1, U) ;
   si T[m1] < T[m2] : devolver m1 ;
   else :             devolver m2 ;
```

Estimar razonadamente el trabajo del algoritmo sobre una tabla de $U - P + 1 = N$ elementos. Para ello, definir una operación básica adecuada, establecer una ecuación para dicho trabajo, resolverla en algún caso particular adecuado y, finalmente, dar la solución general.

9

(1 cara)

Metodología y Tecnología de la Programación II, Febrero 2002

Apellidos: **Nombre:**
Grupo:

1	2	3	4	5	6	T

NOTA IMPORTANTE: NO SE TENDRÁN EN CUENTA RESPUESTAS NO RAZONADAS

1. a. ¿Qué se entiende por algoritmo de ordenación local? Dar una cota inferior de su rendimiento en los casos peor y medio.
   b. Si T_1, T_2, f son funciones positivas crecientes tales que $T_1 \simeq f$ y $T_2 = o(f)$, deducir razonadamente la verdad o falsedad de las siguientes afirmaciones: i. $T_1 = f + O(T_2)$. ii. $T_1 T_2 = o(f)$ iii. $T_1 T_2 = o(f^2)$. iv. $T_2/T_1 = o(1)$.
   c. Dar la evolución del algoritmo ShellSort sobre la tabla `[9 8 7 6 5 4 3 2 1]` suponiendo incrementos de orden 5, 2 y 1. Para los incrementos de órdenes 5 y 2, ordenar directamente las subtablas. Para el incremento de orden 1 aplicar el método de la burbuja y dar su evolución, indicando para cada iteración la evolución de los elementos usados como burbuja.

2. a. Estimar el crecimiento de una función T que cumple $T(N) \leq N^2 + T(\lfloor N/2 \rfloor)/2$ con $T(1) = 0$.
   b. Dar la evolución de la rutina `Partir` sobre la permutación `[9 1 12 4 11 10 3]` suponiendo que la subrutina `Medio` devuelve el primer índice e indicando la posición en cada paso de los índices relevantes. A continuación, dar la subsiguiente evolución del algoritmo QuickSort.

3. a. Dar la relación existente entre los casos peor y medio de un algoritmo de ordenación por cdcs y su árbol de decisión. Deducir de esta relación una cota inferior para el caso peor.
   b. Si vemos la creación de ABdBs como un algoritmo de ordenación, el mismo compara en primer lugar el elemento $T[2]$ con $T[1]$ y luego $T[3]$ con $T[1]$. Suponiendo que $T[1] < T[2]$, dar razonadamente la parte restante del árbol de decisión de dicho método para permutaciones de 4 elementos.

4. a. ¿Cuál es el número mínimo de nodos de un AVL con profundidad P? ¿Cuál es la estructura de un tal árbol?
   b. Dar ejemplos de AVL de profundidad 3, 4 y 5 con un número mínimo de nodos.
   c. Construir razonadamente el árbol AVL asociado a la lista `[15 17 8 6 5 2 14 10 11]`.

5. a. Se quiere construir una tabla hash con direccionamiento abierto y sondeos aleatorios con 1000 posiciones. ¿Cuál es el máximo número de datos a almacenar para que el número medio de sondeos en búsquedas con y sin éxito sea a lo sumo 4?
   b. De un cierto método hash H con direccionamiento abierto se sabe que $A_H^f(N, M) = 1/(1-\lambda)^2$. Hallar razonadamente el valor de $A_H^e(N, M)$.

6. a. Dar una estimación del rendimiento del algoritmo RadixSort cuando se usa sobre una tabla de N elementos cada uno de los cuales está formado por K símbolos de un alfabeto de S caracteres. Si el mismo se usa para tablas con elementos repetidos, ¿para qué tamaños de tabla será más eficaz que un algoritmo óptimo de ordenación por comparación de claves?
   b. Analizar el rendimiento de la siguiente versión recursiva del algoritmo de ordenación por Selección usando las cdc como OB y suponiendo que la rutina `ind MinMalo(tabla T, ind P, ind U)` para la obtención del índice del elemento mínimo de una tabla satisface $n_{MinMalo}(T, P, U) \leq (U - P)^2$:

```
SelectSortRec(tabla T, ind P, ind U)
   si P == U : volver ;
   m = MinMalo(T, P, U) ;
   swap(T[P], T[m]) ;
   SelectSortRec(T, P+1, U) ;
```

10

1. a) Valor: 2 puntos
Un algoritmo de ordenación local es todo aquel algoritmo de ordenación que por cada operación básica (comparación de claves) deshace a lo sumo una inversión.
$W_{local}(N) <= N^2/2 - N/2$
$A_{local}(N) = N^2/4 + O(N)$

1. b) Valor: 4 puntos

i) Falso. Contraejemplo: $T_1 = N^3$; $f = N^3 + N^2$; $T_2 = N$.
ii) Falso. Contraejemplo: $T_1 = N^3$; $f = N^3 + N$; $T_2 = N$.
iii) Verdadero. Se demuestra por la definición de o(f), es decir:

$$\lim_{N \to \infty} \frac{T_1 \cdot T_2}{f^2} = 0$$

iv) Verdadero. Se demuestra por la definición de o(f), es decir:

$$\lim_{N \to \infty} \frac{T_2 / T_1}{1} = 0$$

1. c) Valor: 4 puntos

[9 8 7 6 5 4 3 2 1]

K=5

[4 3 2 1 5 9 8 7 6]

K=2

[2 1 4 3 5 7 6 9 8]

K=1, se aplica método de la burbuja

Primera iteración, burbuja = 2; al final queda
[1 2 4 3 5 7 6 9 8]
Segunda iteración, burbuja = 4; al final queda
[1 2 3 4 5 7 6 9 8]
Tercera iteración, burbuja = 7; al final queda
[1 2 3 4 5 6 7 9 8]
Cuarta iteración, burbuja = 9; al final queda
[1 2 3 4 5 7 6 8 9]

2. a) Valor: 5 puntos
Primero se estima el crecimiento de la función para el caso especial en que $N=2^k$.
Desarrollando, se obtiene que $T(N) <= N^2 + N^2/2^3 + N^2/2^6 + ... + T(N/2^k)/2^k$.
Como $N=2^k$, este último termino es = 0. Entonces queda:

$$T(N) <= N^2 \cdot \sum_{i=0}^{k-1} (1/2^{3i}) = N^2 \cdot \sum_{i=0}^{k-1} (1/2^3)^i$$

Esta suma se resuelve aplicando la fórmula para progresiones geométricas de razón (1/23), que da como resultado

$$T(N) <= 8/7 \,(N^2 - N^{-1}) = 8\,N^2/7 + O(N).$$

Luego se demuestra por inducción que esto se cumple para cualquier N. Para ello suponemos que es cierto para cualquier N', tal que N' < N.
Entonces vemos que si N es par, $\lfloor N/2 \rfloor = N/2$, que es menor que N. Luego

$$T(N) \underset{\text{por definición}}{<=} N^2 + T(\lfloor N/2 \rfloor)/2 \underset{\text{por hipótesis inductiva}}{<=} N^2 + 1/2\,[8/7 \cdot (N/2)^2 + O(N)]$$

UNICAMENTE PARA USO ESCOLAR (IMPRESO EN PAPEL RECICLADO)

UAM UNIVERSIDAD AUTONOMA DE MADRID

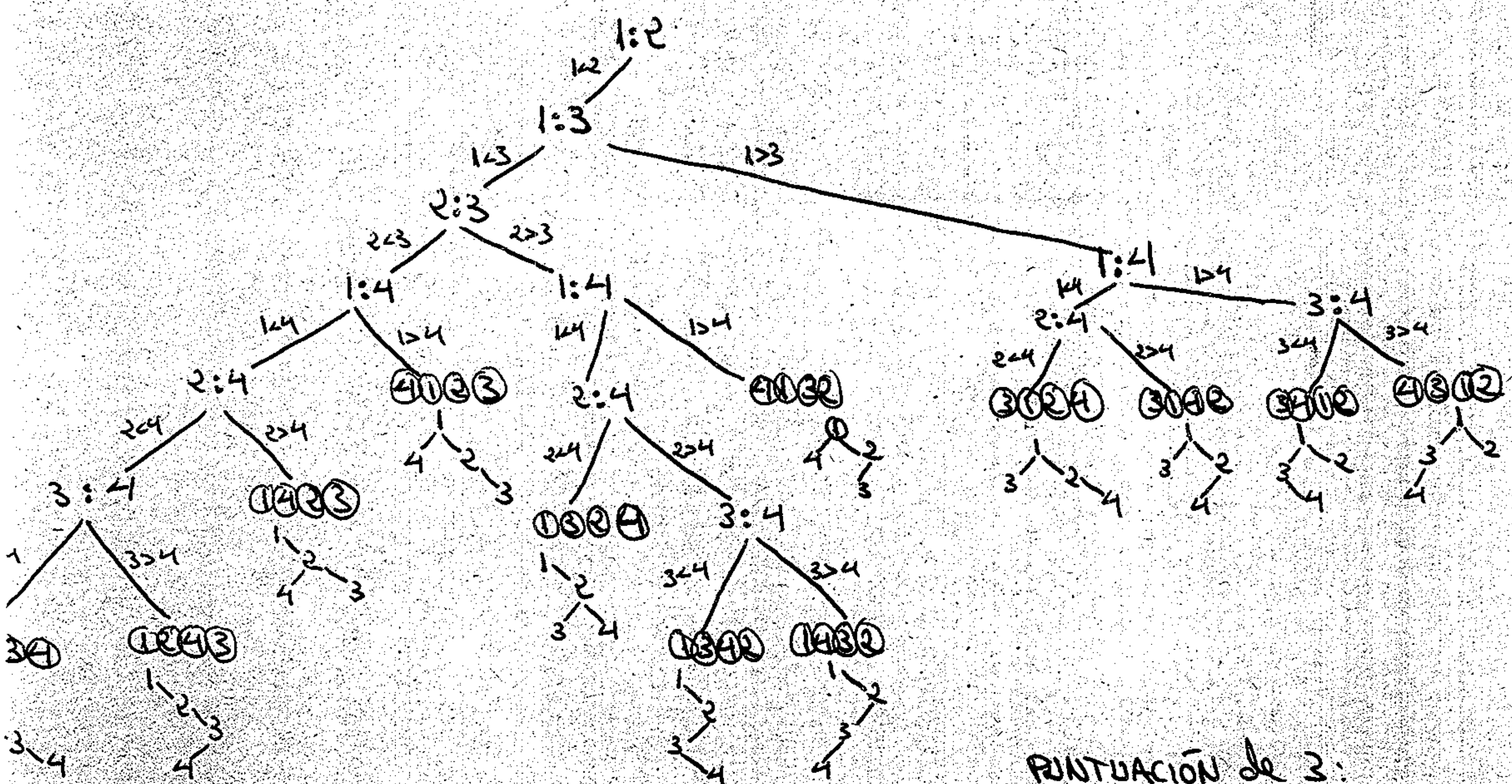
Hoja nº ....

Asignatura .... Grupo ....

Apellidos .... Nombre ....

Ejercicio del día ....

3b



PUNTUACIÓN de 3:

3a: 4 ptos (teoría)

3b: 6 ptos (-1 pto por cada nodo incorrecto) operando, se tiene que $T(N) <= 8 N^2/7 + O(N)$.

De la misma forma, cuando N es impar, $\lfloor N/2 \rfloor = (N-1)/2$, que es menor que N. Luego

$$T(N) \underbrace{<=}_{\text{por definición}} N^2 + T(\lfloor N/2 \rfloor)/2 \underbrace{<=}_{\text{por hipótesis inductiva}} N^2 + ½ [8/7 . ((N-1)/2)^2 + O(N)]$$

operando, se tiene que $T(N) <= 8 N^2/7 + O(N)$, con lo que queda demostrado.

2. b) Valor: 5 puntos

Suponemos que los índices de la tabla están entre 0 y 6. La función *Partir* usará como pivote el elemento de la primera posición, que en la primera invocación será la posición 0, que contiene un 9. Como resultado, la tabla quedará así: [3 1 4 9 11 10 12]; la función *Partir* retornará el índice del 9 en la nueva tabla, 3.

Luego de esto, la función *QuickSort* se invocará recursivamente de la siguiente forma:

QuickSort ([3 1 4 9 11 10 12], 0, 2)

    Partir usara como pivote el 3

QuickSort ([1 3 4 9 11 10 12], 0, 0)

QuickSort ([1 3 4 9 11 10 12], 2, 2)

QuickSort ([1 3 4 9 11 10 12], 4, 6)

    Partir usara como pivote el 11

QuickSort ([1 3 4 9 10 11 12], 4, 4)

QuickSort ([1 3 4 9 10 11 12], 6, 6)

con lo que la tabla queda finalmente ordenada.

Hoja nº ....

UNIVERSIDAD AUTONOMA DE MADRID

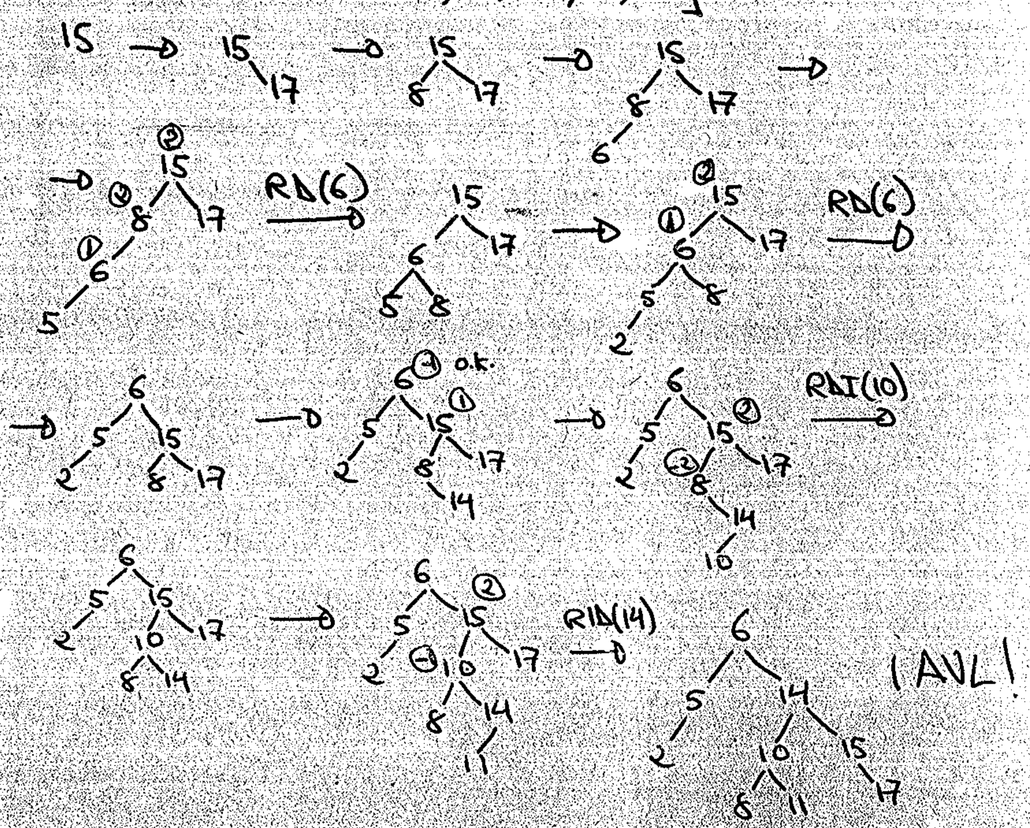
Asignatura .... Grupo ....
Apellidos .... Nombre ....
Ejercicio del día ....

PUNTUACIÓN 4:

4a: 3 ptos } Teoría
4b: 3 ptos

4c: 4 ptos. No se indican rotaciones → -1 pto
Correcto hasta la 2ª rotación → 2 ptos

4c: [15, 17, 8, 6, 5, 2, 14, 10, 11]



#5 ⓐ como $A^e_{SA}(N,m) \leq A^f_{SA}(N,m)$

2.5 basta encontrar el nº. máx de elementos

\+ para que $A^f_{SA}(N, 1000) \leq 4$

2.5 Por tanto

$$A^f_{SA}(N, 1000) = \frac{1}{1-\frac{N}{1000}} \leq 4$$

$$\Rightarrow \frac{1}{4} \leq 1 - \frac{N}{1000} \Rightarrow \frac{N}{1000} \leq 3/4 \Rightarrow N \leq 750$$

ⓑ Se tiene que $A^e(N,m) \simeq \frac{1}{N}\sum_{0}^{N-1} \frac{1}{(1-\frac{i}{m})^2}$

5

$$\simeq \frac{1}{N}\int_0^N \frac{dx}{(1-\frac{x}{m})^2} = \frac{m}{N}\int_0^{N/m} \frac{du}{(1-u)^2}$$

$$= \frac{1}{\lambda}\int_0^{\lambda} \frac{du}{(1-u)^2} = \frac{1}{\lambda}\left[\frac{1}{1-u}\right]_0^{\lambda} = \frac{1}{\lambda}\left(\frac{1}{1-\lambda} - 1\right) = \frac{1}{1-\lambda}$$

#6. ⓐ el coste de RadixSort es esencialmente

1 $O(KN)$ mientras que el coste de

\+

3 un alg. óptimo de ordenación es $O(N \lg N)$

Por tanto, RSort es mejor si $K = O(\lg N)$

esto es si $N = \Omega(2^K)$

ⓑ Si $T(N)$ = nº. de cdc de SSR sobre una

2 tabla de N elementos, se tiene si $U-P+1=N$

\+ $T(N) = n_{\text{mínimo}}(T,P,U) + T(N-1)$

4

$$\leq (N-1)^2 + T(N-1)$$

y $T(1) = 0$.

Desplegando se tiene

$$T(N) \leq (N-1)^2 + (N-2)^2 + T(N-2) \leq \ldots \leq \sum_1^{N-1} i^2 + \underbrace{T(1)}_{=0}$$

$$= \sum_1^{N-1} i^2$$

$$= \frac{N^3}{3} + O(N^2)$$

Metod. y Tecnol. de la Programación II, Noviembre 2002

Apellidos: **Nombre:**
Grupo:

1	2	3	T

NOTA IMPORTANTE: sólo se tendrán en cuenta aquellas respuestas debidamente razonadas
OTRAS OBSERVACIONES Y ADVERTENCIAS: leánse detenidamente antes del inicio del examen

1. El alumno escribirá su nombre en **TODAS** las hojas de examen que se le entreguen y deberá entregar **TODAS ELLAS** al terminar el examen, separando las hojas a corregir de los borradores.

   El no hacerlo así se considerará como indicio de posible participación en copia.

2. Se recuerda que, como es obvio, el alumno **TIENE LA OBLIGACIÓN** de custodiar **ACTIVAMENTE** las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes.

   El no hacerlo así se considerará como indicio de participación en copia.

3. De detectarse casos de copia, los mismos supondrán de entrada el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.

4. Las incidencias de copia detectadas durante el examen o en su corrección se pondrán en conocimiento de la Dirección de la ETS de Informática, así como del resto de los profesores de otras asignaturas en las que estén matriculado los implicados.

Cuestiones

1. a. De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es

$$P(K = T[i]) = \frac{1}{C_N}\frac{1}{\sqrt{i}},$$

donde C_N es una constante que garantiza que $\sum_1^N P(K = T[i]) = 1$.

Calcular razonadamente y con la mayor precisión posible el coste medio de las búsquedas lineales realizadas con éxito.

b. Definir con precisión qué se entiende por las notaciones $f \sim g$, $f = o(g)$, $f = O(g)$ y $f = g + O(h)$.

2. a. Estimar razonadamente y con detalle suficiente el crecimiento de una función T que cumple

$$T(N) \leq 4T(\lfloor N/2 \rfloor) + \sqrt{N}; \quad T(1) = 0.$$

b. Dar la evolución del algoritmo ShellSort sobre la tabla inferior para incrementos 3, 2 y 1:
`[6 10 9 3 4 1 7 5 2]`
Efectuar por inspección las correspondientes ordenaciones y para el incremento 1 indicar razonadamente las cdc a efectuar en la correspondiente ordenación.

3. a. Una posible operación básica en QuickSort es el intercambio de elementos en la rutina `Partir`. ¿Cuáles son los valores de $W_{Partir}(N)$ y $B_{Partir}(N)$? Justificar en suficiente detalle el valor de $W_{QuickSort}(N)$ para dicha OB.

Nota: suponer que `Medio(T, P, U)` devuelve siempre P y contar sólo los intercambios que se hacen en el bucle de `Partir`.

b. ¿Qué se entiende por algoritmo local? Dar 2 ejemplos de tales algoritmos. ¿Cuántas cdc efectuará como mínimo un algoritmo local cualquiera para ordenar la tabla de $N = 2K$ elementos `[K+1 K+2 ... 2K K K-1 ...1]`?

13

Nº 2032

0, 24

Metod. y Tecnol. de la Programación II, Noviembre 2002

Apellidos:
Grupo:
Nombre:

1	2	3	T

NOTA IMPORTANTE: sólo se tendrán en cuenta aquellas respuestas debidamente razonadas
OTRAS OBSERVACIONES Y ADVERTENCIAS: leánse detenidamente antes del inicio del examen

1. El alumno escribirá su nombre en **TODAS** las hojas de examen que se le entreguen y deberá entregar **TODAS ELLAS** al terminar el examen, separando las hojas a corregir de los borradores.

   El no hacerlo así se considerará como indicio de posible participación en copia.

2. Se recuerda que, como es obvio, el alumno **TIENE LA OBLIGACIÓN** de custodiar **ACTIVAMENTE** las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes.

   El no hacerlo así se considerará como indicio de participación en copia.

3. De detectarse casos de copia, los mismos supondrán de entrada el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.

4. Las incidencias de copia detectadas durante el examen o en su corrección se pondrán en conocimiento de la Dirección de la ETS de Informática, así como del resto de los profesores de otras asignaturas en las que estén matriculado los implicados.

Cuestiones

1. a. De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es

$$P(K = T[i]) = \frac{1}{C_N}\frac{1}{\sqrt{i}},$$

   donde C_N es una constante que garantiza que $\sum_1^N P(K = T[i]) = 1$.

   Calcular razonadamente y con la mayor precisión posible el coste medio de las búsquedas lineales realizadas con éxito.

   b. Definir con precisión qué se entiende por las notaciones $f \sim g$, $f = o(g)$, $f = O(g)$ y $f = g + O(h)$.

2. a. Estimar razonadamente y con detalle suficiente el crecimiento de una función T que cumple

$$T(N) \leq 4T(\lfloor N/2 \rfloor) + \sqrt{N}; \quad T(1) = 0.$$

   b. Dar la evolución del algoritmo ShellSort sobre la tabla inferior para incrementos 3, 2 y 1:
   `[6 10 9 3 4 1 7 5 2]`
   Efectuar por inspección las correspondientes ordenaciones y para el incremento 1 indicar razonadamente las cdc a efectuar en la correspondiente ordenación.

3. a. Una posible operación básica en QuickSort es el intercambio de elementos en la rutina `Partir`. ¿Cuáles son los valores de $W_{Partir}(N)$ y $B_{Partir}(N)$? Justificar en suficiente detalle el valor de $W_{QuickSort}(N)$ para dicha OB.

   Nota: suponer que `Medio(T, P, U)` devuelve siempre P y contar sólo los intercambios que se hacen en el bucle de `Partir`.

   b. ¿Qué se entiende por algoritmo local? Dar 2 ejemplos de tales algoritmos. ¿Cuántas cdc efectuará como mínimo un algoritmo local cualquiera para ordenar la tabla de $N = 2K$ elementos
   `[K+1 K+2 ... 2K K K-1 ...1]`?

1,4

1

#1.a

Si $A^{e}_{BL}(N)$ = nº medio de cdcs en búsqueda con éxito, se tiene

$$A^{e}_{BL}(N) = A(N) = \sum_{i=1}^{N} \frac{1}{C_N} \frac{i}{\sqrt{i}} \quad (i)$$

$$= S_N / C_N,$$

donde

$$S_N = \sum_{1}^{N} i^{1/2}, \quad C_N = \sum_{1}^{N} i^{-1/2}$$

Estimamos cada una por integración

Para S_N como $x^{1/2} \uparrow$ (ii), se tiene

$$\int_0^N x^{1/2}\,dx \le S_N \le \int_1^{N+1} x^{1/2}\,dx \quad (iii)$$

$$\Rightarrow_{(iv)} \frac{2}{3} N^{3/2} \le S_N \le \frac{2}{3}(N+1)^{3/2} - \frac{2}{3}$$

$$\Rightarrow 1 \le \underbrace{\frac{S_N}{\frac{2}{3} N^{3/2}}}_{(v)} \le \underbrace{\left(\frac{N+1}{N}\right)^{3/2}}_{\xrightarrow[N\to\infty]{} 1} - \underbrace{\frac{1}{N^{3/2}}}_{\xrightarrow[N\to\infty]{} 0}$$

$$\Rightarrow S_N \sim \frac{2}{3} N^{3/2}$$

(vi) Análogamente, como $1/\sqrt{x} \downarrow$ se tiene (vii)

$$\int_0^N x^{-1/2}\,dx \ge C_N \ge \int_1^{N+1} x^{-1/2}\,dx$$

y razonando como antes se llega a

$$C_N \simeq 2 N^{1/2}$$

Por tanto $A(N) = f_N / C_N \underset{(2*)}{\sim} \dfrac{\frac{2}{3} N^{3/2}}{2 N^{1/2}} = \dfrac{N}{3}$ ②,

ya que

$$\frac{A(N)}{N/3} = \underbrace{\frac{f_N}{\frac{2}{3} N^{3/2}}}_{\underset{N\to\infty}{\longrightarrow} 1} \cdot \frac{1}{\underbrace{\frac{C_N}{2N^{1/2}}}_{\underset{N\to\infty}{\longrightarrow} 1}} \underset{N\to\infty}{\longrightarrow} 1$$

(*)

2. a Estimamos $T(N)$ en 2 pasos

(i) solución particular para $N = 2^K$ ①

$$T(N) \leq \sqrt{N} + 4T\left(\frac{N}{2}\right) \leq \sqrt{N} + 4\left(\sqrt{\frac{N}{2}} + 4T\left(\frac{N}{2^2}\right)\right)$$

$$= \sqrt{N}\left(1 + \frac{4}{\sqrt{2}}\right) + 4^2 T\left(N/2^2\right)$$

$$\leq \sqrt{N}\left(1 + \frac{4}{\sqrt{2}}\right) + 4^2\left[\sqrt{\frac{N}{2^2}} + 4T\left(N/2^3\right)\right]$$

$$= \sqrt{N}\left(1 + \frac{4}{\sqrt{2}} + \left(\frac{4}{\sqrt{2}}\right)^2\right) + 4^3 T\left(N/2^3\right)$$

$$\leq \cdots \underset{②}{\leq} \sqrt{N} \sum_{j=0}^{K-1} \left(4/\sqrt{2}\right)^j + \underbrace{4^K T(\underbrace{N/2^K}_{=1})}_{=0}$$

$$\underset{③}{=} \sqrt{N}\, \frac{(2\sqrt{2})^K - 1}{2\sqrt{2} - 1} =$$

$$= \sqrt{N}\, \frac{(2^K)^{3/2} - 1}{2\sqrt{2} - 1} = \frac{\sqrt{N}\,(N^{3/2} - 1)}{2\sqrt{2} - 1}$$

$$\underset{④}{=} \frac{1}{2\sqrt{2} - 1}\left(N^2 - \sqrt{N}\right)$$

3

(iii) Comprobamos por inducción (5) que $\forall N$, $T(N) \leq \frac{1}{2\sqrt{2}-1} \underbrace{(N^2 - \sqrt{N})}_{f(N) \uparrow}$

Se tiene

$$T(N) \leq \sqrt{N} + 4\, T\left(\left\lfloor \frac{N}{2} \right\rfloor\right)$$

(6) HI

$$\leq \sqrt{N} + 4 \frac{1}{2\sqrt{2}-1}\left(\left\lfloor\frac{N}{2}\right\rfloor^2 - \sqrt{\left\lfloor\frac{N}{2}\right\rfloor}\right)$$

$f(N)\uparrow$

(7) $$\leq \sqrt{N} + 4\frac{1}{2\sqrt{2}-1}\left(\frac{N^2}{4} - \frac{\sqrt{N}}{\sqrt{2}}\right)$$

$$= \frac{1}{2\sqrt{2}-1} N^2 + \sqrt{N}\left(1 - \frac{2\sqrt{2}}{2\sqrt{2}-1}\right)$$

(8) $$= \frac{1}{2\sqrt{2}-1}\left(N^2 - \sqrt{N}\right) \checkmark$$

#2. (i) $unc = 3$ 6 3 7

10 4 5

9 1 2

$\Rightarrow$ 3 4 1 6 5 2 7 10 9

(ii) $uncr = 2$ 3 1 5 7 9

4 6 2 10

$\Rightarrow$ 1 2 3 4 5 6 7 10 9

1 1 1 1 1 1 1 2

(iii) Para uncr 1 y sobre cada elemento se hacen las cdcs indicadas como subíndices. En total 9 cdcs

#3.a claramente con la OB proporciona a tiene

$B_p(N) \geq 0 \Leftarrow 0 \leqslant \kappa_{Partir}(\xi) \leqslant N-1 \Rightarrow W_p(N) \leq N-1$

Además $\kappa_p(1\ 2 \ldots N) = 0$

$\kappa_p(N\ N-1, \ldots 2\ 1) = N-1$

Por tanto

$W_p(N) = N-1$ y $B_p(N) = 0$

Se cumple que si $\sigma(1) = i$

$$n_p(\sigma) = i - 1$$

Ahora, si $\sigma_N = (N \; 1 \; 2 \; \ldots \; N-1)$

$n_p(\sigma_N) = N-1$ y σ_N queda tras partir

como $(\underbrace{N-1 \; 1 \; 2 \; \ldots \; N-2}_{\sigma_{N-1}} \; \boxed{N})$

$\Rightarrow n_p(\sigma_{N-1}) = N-2$ y σ_{N-1} queda

$$(\underbrace{N-2 \; 1 \; 2 \; \ldots \; N-3}_{\sigma_{N-2}} \; \boxed{N-1})$$

como consecuencia

$$n_{QS}(\sigma_N) = N-1 + n_{QS}(\sigma_{N-1})$$

$$= N-1 + N-2 + n_{QS}(\sigma_{N-2}) = \ldots$$

$$= N-1 + N-2 + \ldots + 1 + \underbrace{n_{QS}(\sigma_1)}_{= 0} \quad (1)$$

$$= \sum_{1}^{N-1} i = \frac{(N-1)N}{2}$$

(6)

vemos que en general y por inducción

$$n^{m}_{QS}(\sigma) \leq \frac{N^2}{2} - \frac{N}{2}$$

Sup que $\sigma(1) = i$;

entonces

$$n_{QS}(\sigma) \leq i-1 + W_{QS}(i-1) + W_{QS}(N-i)$$

$$\overset{HI}{\leq} i-1 + \frac{(i-1)^2}{2} - \frac{i-1}{2} + \frac{(N-i)^2}{2} - \frac{N-i}{2}$$

$$= i - 1 + \frac{i^2}{2} - i + \frac{1}{2} - \frac{i}{2} + \frac{1}{2} + \frac{N^2}{2} - Ni + \frac{i^2}{2} - \frac{N}{2} + \frac{i}{2}$$

$$= \frac{N^2}{2} - \frac{N}{2} + i^2 - Ni$$

$$= \frac{N^2}{2} - \frac{N}{2} + \underbrace{i(i-N)}_{\leq 0}$$

$$\leq \frac{N^2}{2} - \frac{N}{2}$$

#3.b Se tiene

$$\sigma = (\underbrace{k+1 \ldots 2k}_{1} \; \underbrace{k \; k-1 \ldots 1}_{2})$$

y si α es local

$$n_\alpha(\sigma) \geq inv(\sigma) = inv \, ① + inv \, ②$$

claramente cada elemento de ① está invertido con todo ②

$$\Rightarrow inv \, ① = k^2$$

Además

$$inv \, ② = \sum_{1}^{k-1} i = \frac{(k-1)k}{2}$$

$$\Rightarrow n_\alpha(\sigma) \geq \frac{3k^2}{2} - \frac{k}{2} = \frac{3}{8} n^2 - \frac{n}{4}$$

2025

0'16

Metod. y Tecnol. de la Programación II, Noviembre 2003

Apellidos: **Nombre:**
Grupo:

1	2	3	T

NOTA IMPORTANTE: sólo se tendrán en cuenta aquellas respuestas debidamente razonadas
OTRAS OBSERVACIONES Y ADVERTENCIAS: léanse detenidamente antes del inicio del examen

1. El alumno escribirá su nombre en **TODAS** las hojas de examen que se le entreguen y deberá entregar **TODAS ELLAS** al terminar el examen, separando las hojas a corregir de los borradores.

   El no hacerlo así se considerará como indicio de posible participación en copia.

2. Se recuerda que, como es obvio, el alumno **TIENE LA OBLIGACIÓN** de custodiar **ACTIVAMENTE** las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes.

   El no hacerlo así se considerará como indicio de participación en copia.

3. De detectarse casos de copia, los mismos supondrán de entrada el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.

4. Las incidencias de copia detectadas durante el examen o en su corrección se pondrán en conocimiento de la Dirección de la ETS de Informática, así como del resto de los profesores de otras asignaturas en las que estén matriculado los implicados.

Cuestiones

1. a. Definir qué se entiende por heap y por max–heap.

   b. El algoritmo RadixSort ordena una tabla con N K–uplas de un alfabeto Σ. ¿Qué parámetros determinan su rendimiento? ¿Cuál es el mismo? ¿En qué condiciones su rendimiento NO es mejor que el de, por ejemplo, HeapSort?

   c. Si T_1, T_2, f son funciones positivas crecientes con

$$lim_{N\to\infty} T_1(N) = lim_{N\to\infty} T_2(N) = lim_{N\to\infty} f(N) = \infty$$

   y tales que

$$T_1 \simeq f, T_2 = o(f),$$

   deducir razonadamente la verdad o falsedad de las siguientes afirmaciones:
   i. $T_2 = o(T_1)$. ii. $e^{T_1} \simeq e^f$
   iii. $\log T_1 = \log f + o(1)$. iv. $T_1/T_2 = o(f)$.

18

2. a. Dar la evolución del algoritmo HeapSort sobre la tabla [25 3 16 15 38 2 21] usando para ello su representación como árbol binario.

   b. ¿Qué se entiende por algoritmo local? ¿Cuántas cdc efectuará como mínimo un algoritmo local $\mathcal{A}$ cualquiera para ordenar la tabla σ de $N = 4K$ elementos
   [K+1 K+2 ... 2K 3K 3K-1 ... 2K+1 4K 4K-1 ... 3K+1 K K-1 ...1]?

3. a. Dada una tabla T, el algoritmo InsertSort efectúa en primer lugar la comparación entre $T[1]$ y $T[2]$. Suponiendo que $T[1] > T[2]$, ¿contra qué elemento $T[j]$ se compara entonces $T[1]$?

   b. Suponiendo que $T[1]$ gana dicha comparación, dar razonadamente el subárbol de decisión que sigue a la misma para tablas de 4 elementos (esto es, el subárbol que sigue por la derecha a la raíz es $1 : 2$, y a su hijo derecho de la forma $1 : j$). Tras cada cdc, indicar el estado de la tabla y marcar el elemento a insertar.

1. c

i. cierto, pues $\frac{T_2}{T_1} = \frac{T_2}{f} \cdot \frac{1}{\frac{T_1}{f}} \to 1$

$\frac{T_2}{f} \to 0$, $\frac{T_1}{f} \to 1$

ii. falso: si $T_1 = N^2 + N$, $f = N^2$ entonces $T_1 \sim f$, pero $e^{T_1} = e^{N^2} \cdot e^{N} = e^{f} \cdot e^{N}$

y $e^{f} = o(e^{T_1})$

iii. cierto, pues $|\log T_1 - \log f| =$

$= |\log T_1/f| \to 0$

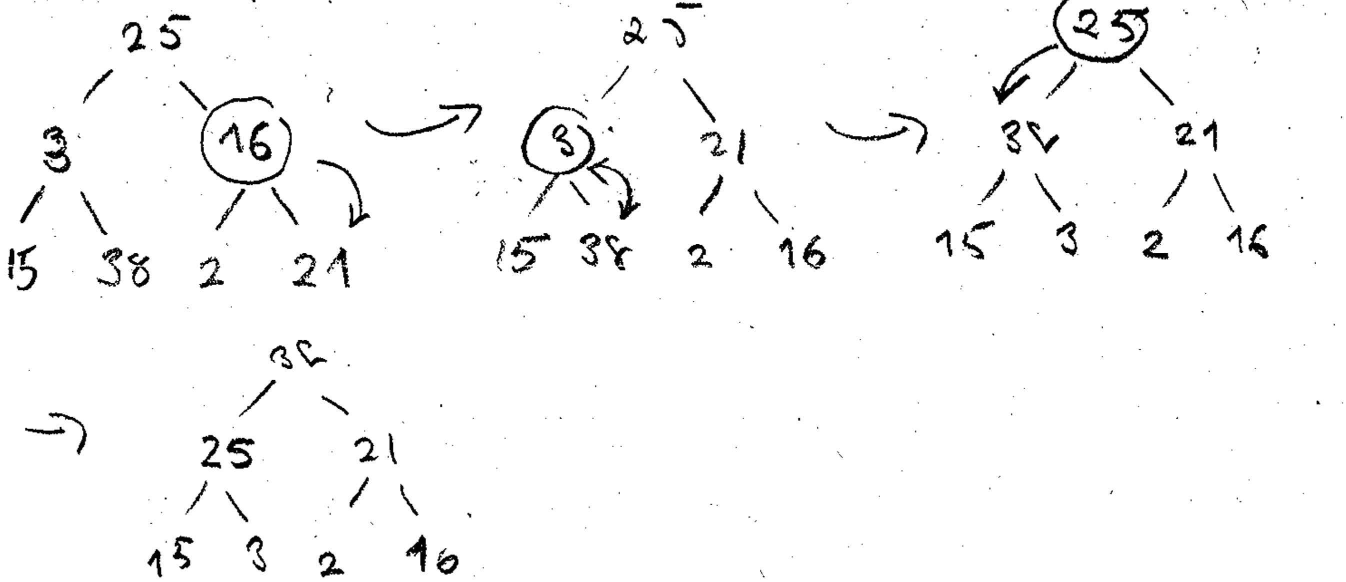
iv. cierto, pues $\frac{T_1/T_2}{f} = \frac{T_1}{f} \cdot \frac{1}{T_2} \to 0$

$\frac{T_1}{f} \to 1$

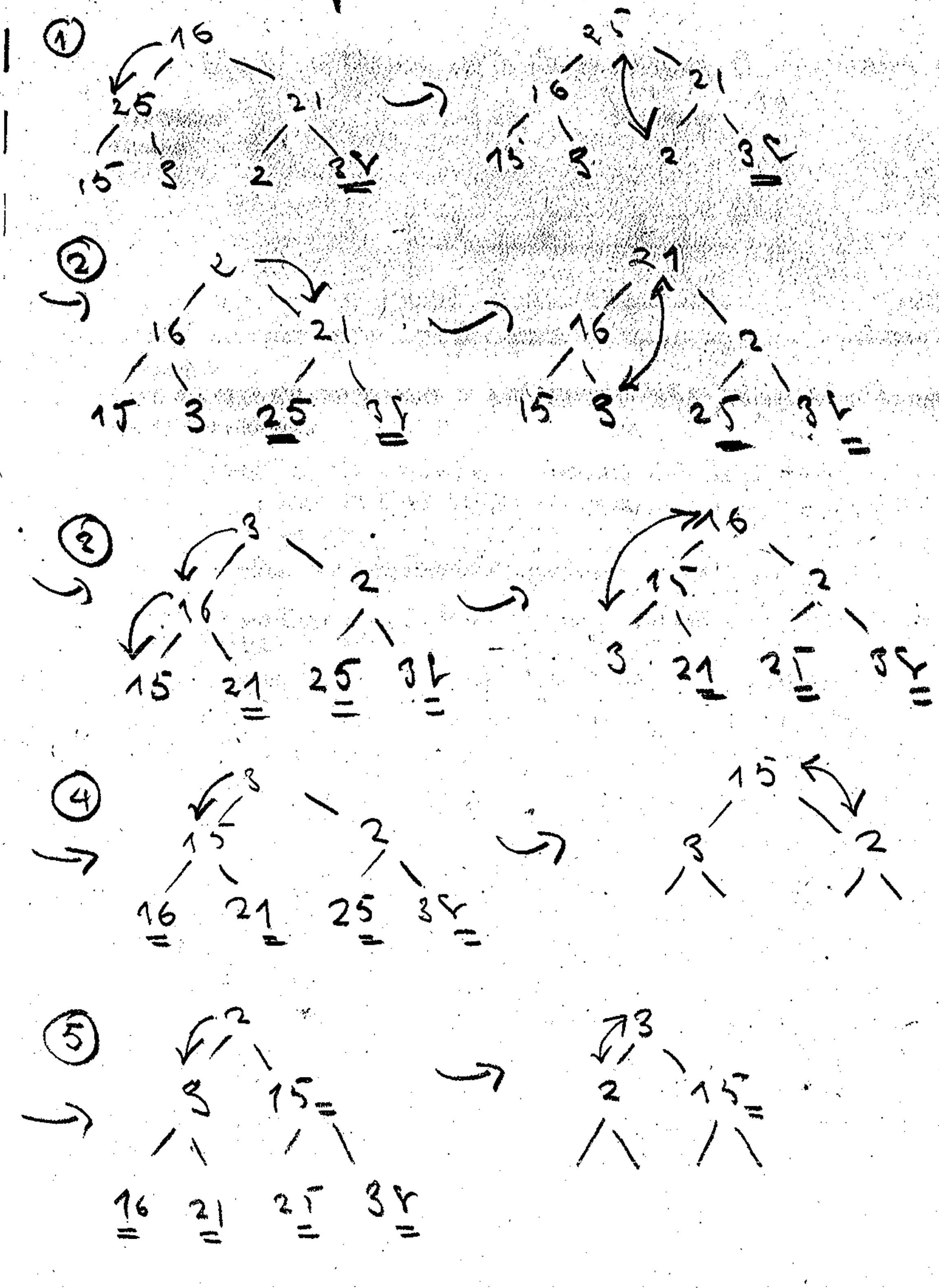
$\frac{1}{T_2} \to 0$ pues $T_2 \uparrow \infty$

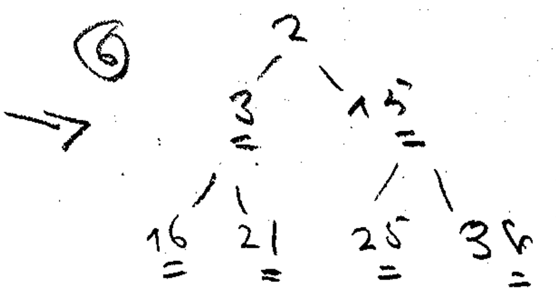
2. a

con Heap

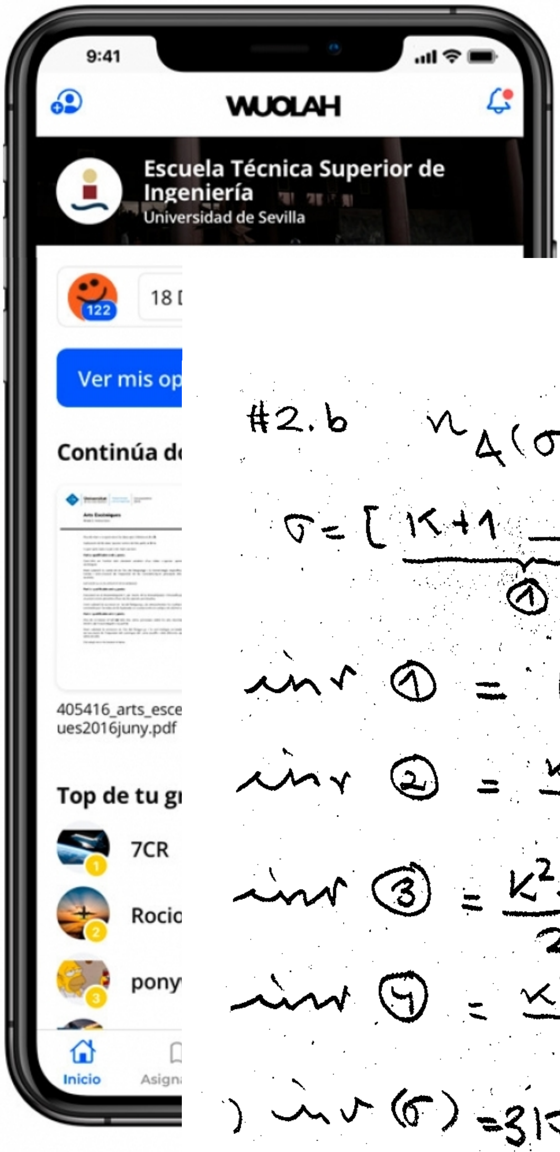


Ordenar Heap





y ya está ordenado



#2.b $n_4(\sigma) \geq inv(\sigma)$

$$\sigma = [\underbrace{k+1 \ldots 2k}_{1} \; \underbrace{3k \ldots 2k+1}_{2} \; \underbrace{4k \ldots 3k+1}_{3} \; \underbrace{k \ldots 1}_{4}]$$

inv ① $= k^2$ (con vegue ④)

inv ② $= \frac{k(k-1)}{2}$ (internas) $+ k^2$ (con el ④)

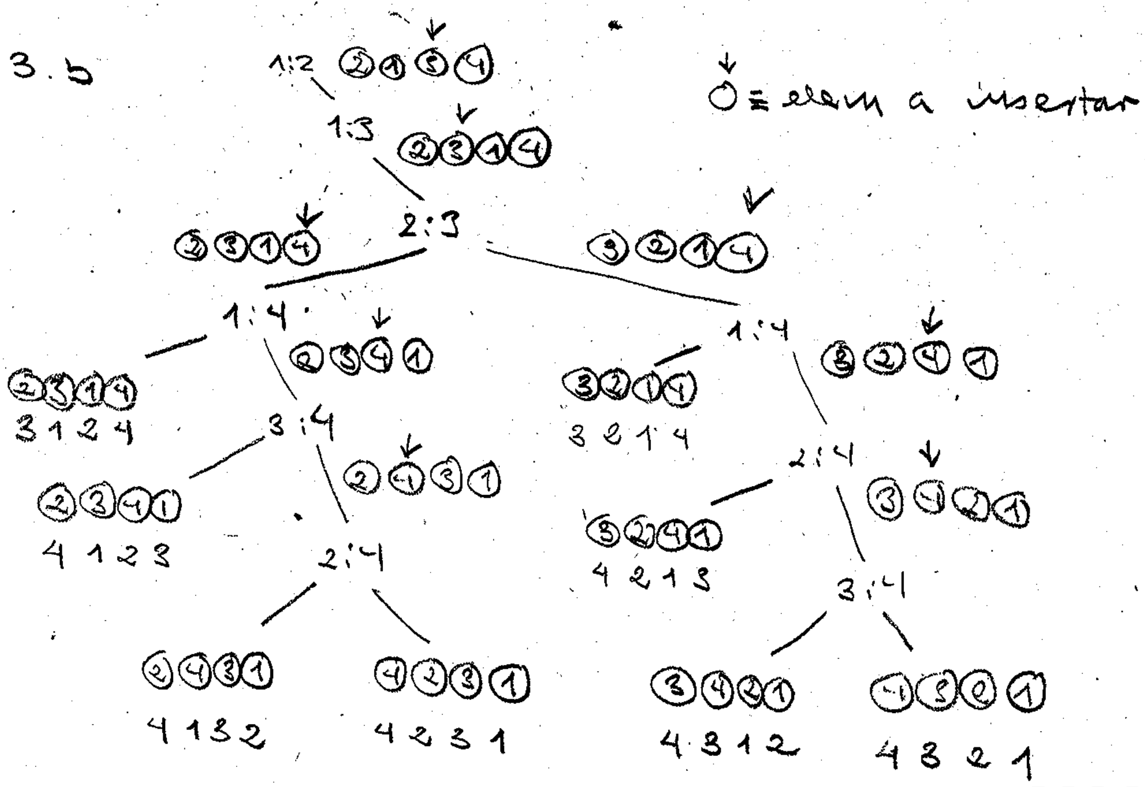
inv ③ $= \frac{k^2-k}{2}$ (internas) $+ k^2$ (con el ④)

inv ④ $= \frac{k^2-k}{2}$ (internas).

$$\Rightarrow inv(\sigma) = 3k^2 + 3\left(\frac{k^2}{2} - \frac{k}{2}\right) = \frac{9k^2}{2} + O(k)$$

$$= \frac{9}{2}\left(\frac{N}{4}\right)^2 + O(N) = \frac{9}{32}N^2 + O(N)$$

3.b



Metod. y Tecnol. de la Programación II, Febrero 2004

Apellidos: **Nombre:**
Grupo: **Aula:** **Bloque:**

1	2	3	4	5	6	T

NOTAS IMPORTANTES:

- **Sólo se tendrán en cuenta aquellas respuestas debidamente razonadas**
- **Se recuerda que según las normas de evaluación para el curso 2003–04, para superar el examen será necesario obtener 7/10 de los puntos de la primera parte. De no ser así, no se corregirán los problemas de la segunda parte.**

OTRAS OBSERVACIONES Y ADVERTENCIAS: léanse detenidamente antes del inicio del examen

1. El alumno escribirá su nombre en **TODAS** las hojas de examen que se le entreguen y deberá entregar **TODAS ELLAS** al terminar el examen, separando las hojas a corregir de los borradores.

   El no hacerlo así se considerará como indicio de posible participación en copia.

2. Se recuerda que, como es obvio, el alumno **TIENE LA OBLIGACIÓN** de custodiar **ACTIVAMENTE** las hojas y otros materiales suyos con los que trabaje en el examen, manteniéndolos fuera del alcance visual o físico de otros estudiantes.

   El no hacerlo así se considerará como indicio de participación en copia.

3. De detectarse casos de copia, los mismos supondrán de entrada el suspenso de todos los implicados, bien sean fuentes o receptores, sin perjuicio de otras medidas disciplinarias que puedan aplicarse.

4. Las incidencias de copia detectadas durante el examen o en su corrección se pondrán en conocimiento de la Dirección de la EPS, así como del resto de los profesores de otras asignaturas en las que estén matriculado los implicados.

Cuestiones de la primera parte

1. a. (1 punto) ¿Cuál es el número mínimo de elementos de un AVL de profundidad P?

   b. (1 punto) Definir con precisión qué se entiende por las notaciones $f \sim g$, $f = o(g)$.

   c. (1 punto) ¿Cuál es el coste medio de las búsquedas con fracaso y con éxito de un hash aleatorio con encadenamiento?

   d. (3 puntos) El algoritmo heapSort ha producido en su fase de construcción del heap el max–heap inferior. Indicar sobre un AB los pasos que dicho algoritmo efectúa en la fase de ordenación sobre este max–heap.

   e. (4 puntos) Indicar la evolución del algoritmo Shell sobre la tabla inferior suponiendo incrementos de valor 4, 3 y 1.

[8 7 6 1 12 3 10 9 4 11 2 5]

Para los incrementos de órdenes 4 y 3, ordenar directamente las subtablas. Para el incremento de orden 1 aplicar el método de inserción y dar su evolución, indicando para cada iteración de este método la ordenación parcial de la tabla en cada momento y evaluando razonadamente el número de cdcs que efecúa el método de inserción sobre esta última tabla.

2. a. (1 punto) ¿Cuál es el rendimiento en número de cdcs del método Quicksort en el caso peor y en el caso medio? Dar un ejemplo de permutación qué produzca el máximo número de cdcs para el Quicksort.

   b. (2 puntos) Dar una relación entre el coste en el caso peor de un algoritmo de ordenación por cdcs y su árbol de decisión. ¿Qué cota inferior se deduce de dicha relación?

   c. (8 puntos) De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es

$$P(K = T[i]) = \frac{1}{C_N}\frac{1}{i^{1/3}},$$

   donde C_N es una constante que garantiza que $\sum_1^N P(K = T[i]) = 1$.

   Calcular razonadamente y con la mayor precisión posible el coste medio de las búsquedas lineales realizadas con éxito.

3. a. (4 puntos) De un cierto método hash con direccionamiento abierto se sabe que $A^f(N,m) = \frac{1}{(1-\lambda)^2}$ y $A^e(N,m) = \frac{1}{1-\lambda}$, donde λ denota el correspondiente factor de carga. Si se quieren almacenar en una tal tabla 1000 datos, estimar razonadamente un tamaño de tabla que garantice que el número medio de búsquedas sea a lo sumo 4.

   b. (5 puntos) Construir el árbol AVL asociado a la lista (2 14 5 13 7 11 8 9).

   c. (1 punto) Sobre el AVL asociado a la lista anterior, y considerándolo sólo como ABdB, dar su estado tras quitar los elementos 7 y 8.

Cuestiones de la segunda parte

4. Estimar razonadamente el crecimiento de una función T que cumple $T(N) \leq N^{1/3} + T(\lfloor N/8 \rfloor)$ con $T(1) = 0$. Se han de seguir para ello los siguientes pasos:
   i. (6 puntos) Dar primero una estimación para un caso particular adecuado.
   ii. (4 puntos) Deducir de ahí una solución para el caso general.

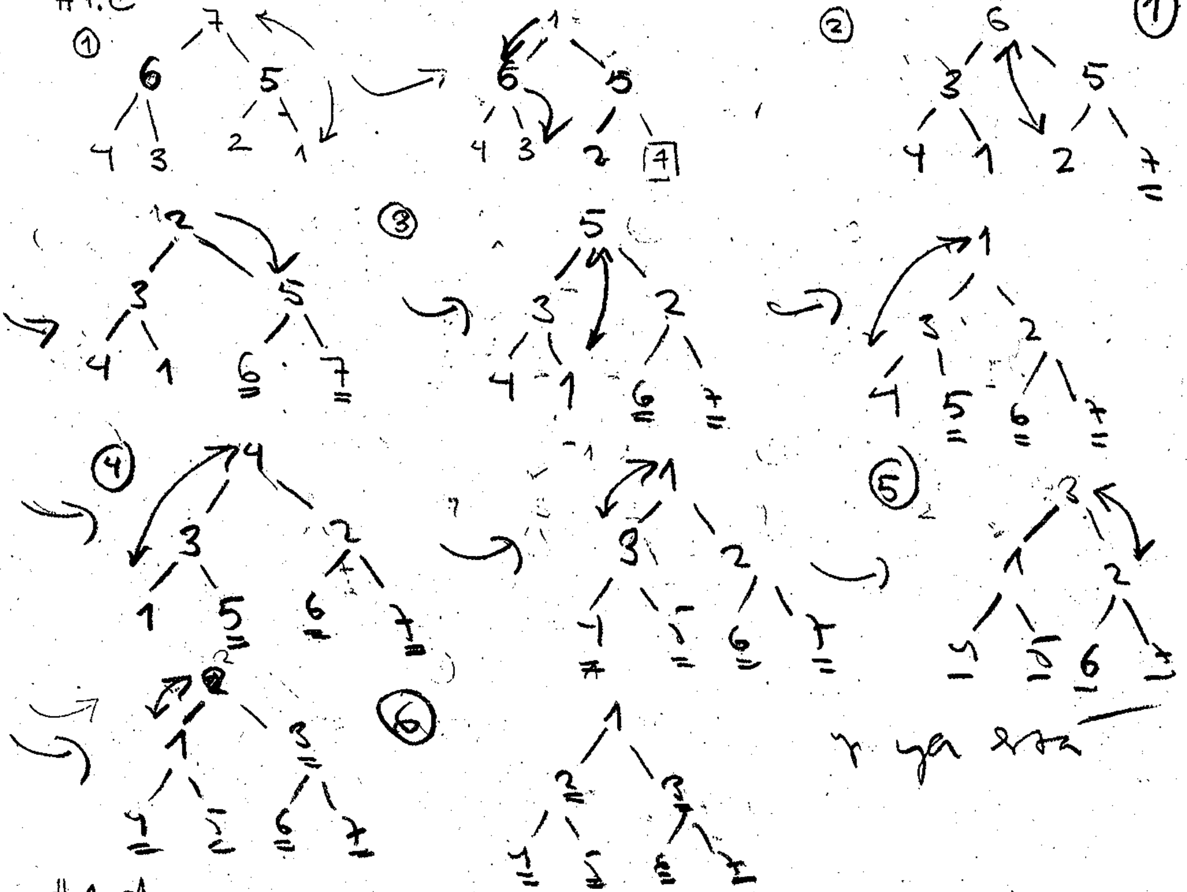
5. a. (2 puntos) Dada una tabla T de 4 elementos, el algoritmo MergeSort efectúa en primer lugar la comparación 1 : 2 entre los elementos 1 y 2 de la tabla inicial. Indicar razonadamente qué comparación $i : j$, con $i < j$, efectúa entonces MergeSort.

   b. (8 puntos) Suponiendo que el elemento 1 es mayor que el 2 y que en la segunda comparación el elemento i es menor que el j, dar razonadamente el subárbol de decisión que sigue a esta segunda comparación para tablas de 4 elementos. Tras cada cdc, indicar el estado de las subtablas implicadas e indicar los elementos a comparar.

6. a. (4 puntos) De un algoritmo hash con direccionamiento abierto se sabe que $A^f(N,m) = 1/(1-\frac{N}{m})^3$. Estimar razonadamente $A^e(N,m)$ explicando brevemente el enfoque usado para dicha estimación.

   b. (6 puntos) Se quiere aplicar el algoritmo de inserción sobre un conjunto de tablas T de tamaño N y valores entre 1 y N, donde sistemáticamente su primera mitad está formada por los enteros de 1 a $N/2$ ya ordenados. Estimar razonadamente en función de N cuál será el coste medio de tales ordenaciones. Estimar también dicho coste medio si la tabla de la primera mitad estuviera formada por los enteros de 1 a $N/2$ en orden decreciente.

#1.c



y ya está

#1.d

(1) k=4
8 7 6 1 | 12 3 10 9 | 4 11 2 5

→ 4 3 2 1 8 7 6 5 12 11 10 9

(2) k=3
4 3 2 | 1 8 7 | 6 5 12 | 11 10 9

→ 1 3 2 4 5 7 6 8 9 11 10 12

(3) 1 3 → 1 3 2 → 1 2 3 4 → 1 2 3 4 5 → 1 2 3 4 5 7 → 1 2 3 4 5 7 6 → … 5 6 7 8 → … 6 7 8 9 → … 8 9 11 → … 9 11 10 → … 9 10 11 12

6. Se tiene ②

$$A^e_{BL}(N) = \sum_1^N \frac{1}{C_N\, i^{1/3}}\, i = \frac{1}{C_N}\sum_1^N i^{2/3} = \frac{S_N}{C_N}$$

con $S_N = \sum_1^N i^{2/3}$, $C_N = \sum_1^N 1/i^{1/3}$

Analizamos S_N. Como $f(x) = x^{2/3} \uparrow$, se tiene

$$\int_0^N x^{2/3}\,dx \leq S_N \leq \int_1^{N+1} x^{2/3}\,dx$$

$$\Rightarrow \left[\frac{3}{5}x^{5/3}\right]_0^N \leq S_N \leq \left[\frac{3}{5}x^{5/3}\right]_1^{N+1}$$

$$\Rightarrow \frac{3}{5}N^{5/3} \leq S_N \leq \frac{3}{5}(N+1)^{5/3} - \frac{3}{5}$$

y se tiene $S_N \sim \frac{3}{5}N^{5/3}$

Análogamente, como $1/x^{1/3} \downarrow$,

$$\int_0^N \frac{dx}{x^{1/3}} \geq C_N \geq \int_1^{N+1} \frac{dx}{x^{1/3}}$$

$$\Rightarrow \frac{3}{2}N^{2/3} \geq C_N \geq \frac{3}{2}(N+1)^{2/3} - \frac{3}{2}$$

y $C_N \sim 3/2\, N^{2/3}$

Por tanto $A^e_{BL}(N) \sim \dfrac{\frac{3}{5}N^{5/3}}{\frac{3}{2}N^{2/3}} = \frac{2}{5}N$, ya que

$$\frac{A^e_{BL}(N)}{\frac{2}{5}N} = \underbrace{\frac{S_N}{\frac{3}{5}N^{5/3}}}_{\to 1} \cdot \frac{1}{\underbrace{\frac{C_N}{\frac{3}{2}N^{2/3}}}_{\to 1}} \to 1$$

③

#3.a se requiere $A^f(N,m) = \frac{1}{(1-\lambda)^2} \leq 4$ pues la otra condición $A^e(N,m) \leq 4$ es automática.

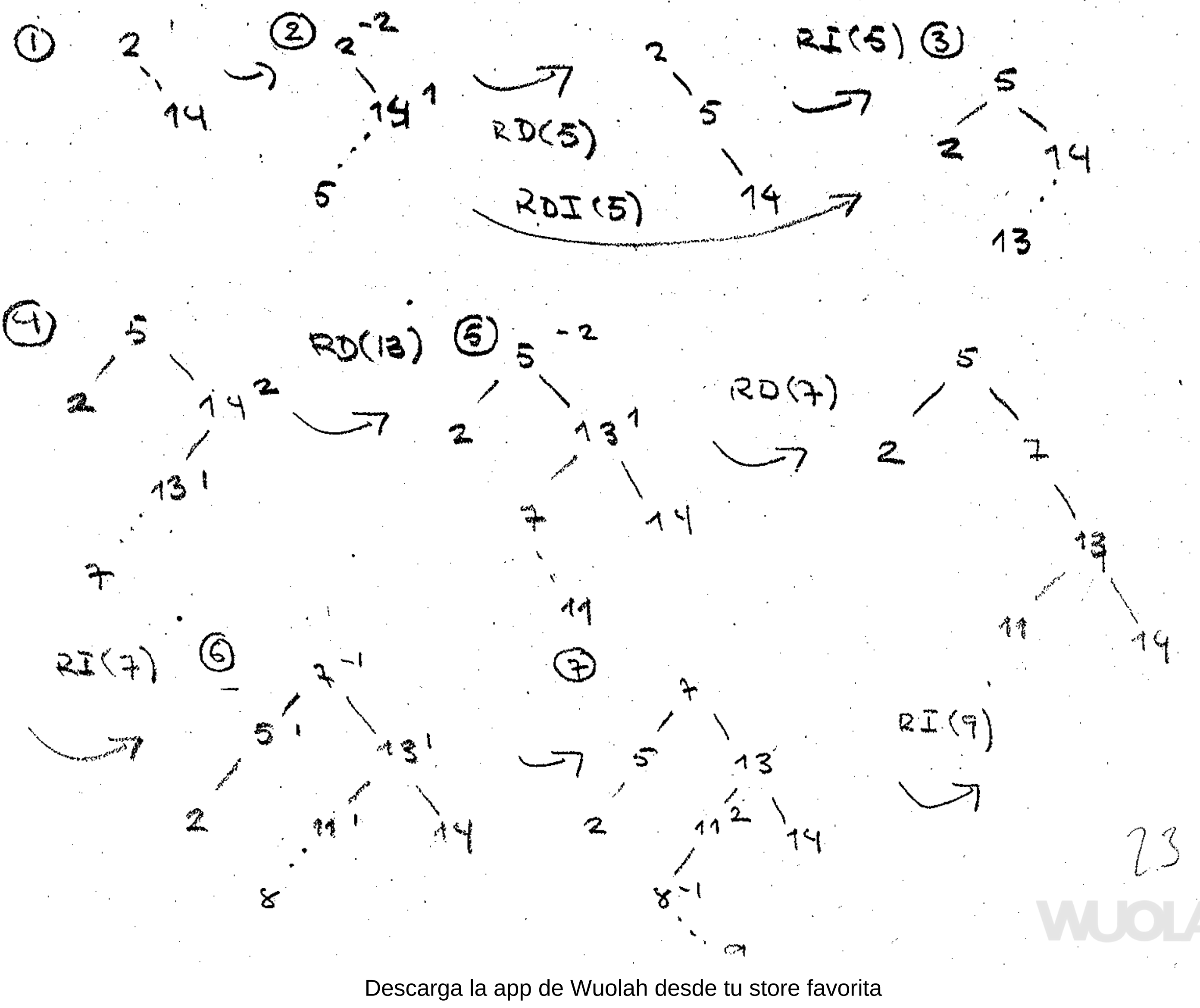
Entonces

$$\frac{1}{4} \leq (1-\lambda)^2 \Rightarrow \frac{1}{2} \leq 1-\lambda \Rightarrow \lambda = \frac{1000}{m} \leq \frac{1}{2}$$

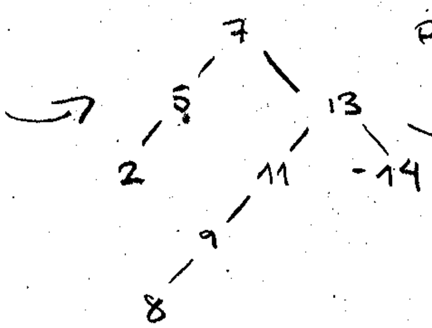
$\Rightarrow$ m ha de ser ≥ 2000

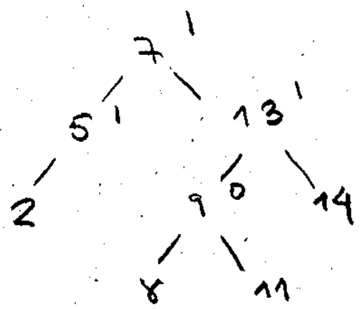
Se observa que si $\lambda \leq 1/2$, $A^e(N,m) = \frac{1}{1-\lambda} \leq \frac{1}{1-1/2} = 2 \leq 4$

#3.b



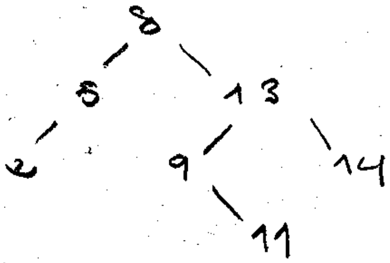
23



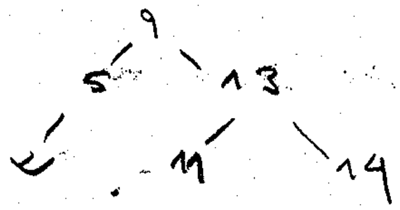


$\Rightarrow$ AVL!

#3.c quitamos 7:
como el sucesor es
el 8 no tiene



quitamos 8:
sucesor(8) = 9



#4. (i) se considera $N = 8^K$

$$T(N) \leq N^{1/3} + \left(\left(\frac{N}{8}\right)^{1/3} + T(N/8^2) \right)$$

$$= N^{1/3}\left(1 + \frac{1}{2}\right) + T(N/8^2) \leq N^{1/3}\left(1 + \frac{1}{2}\right) +$$

$$+ \left(\frac{N^{1/3}}{8^{2/3}} + T(N/8^3) \right)$$

$$= N^{1/3}\left(1 + \frac{1}{2} + \frac{1}{2^2}\right) + T(N/8^3)$$

$$\leq \dots \leq N^{1/3} \sum_{0}^{K-1} 1/2^j + \underbrace{T(\overbrace{N/8^K}^{=1})}_{=0}$$

$$= N^{1/3} \frac{1 - 1/2^K}{1 - 1/2} = 2N^{1/3}\left(1 - \frac{1}{8^{K/3}}\right) =$$

$$= 2(N^{1/3} - 1)$$

ii. caso general: por inducción (5)

C. Base: $T(1) = 0 \leq \frac{1}{2}(1^{1/3} - 1)$

C. General: sup. $\forall N' < N,\ T(N') \leq 2(N'^{1/3} - 1)$

entonces

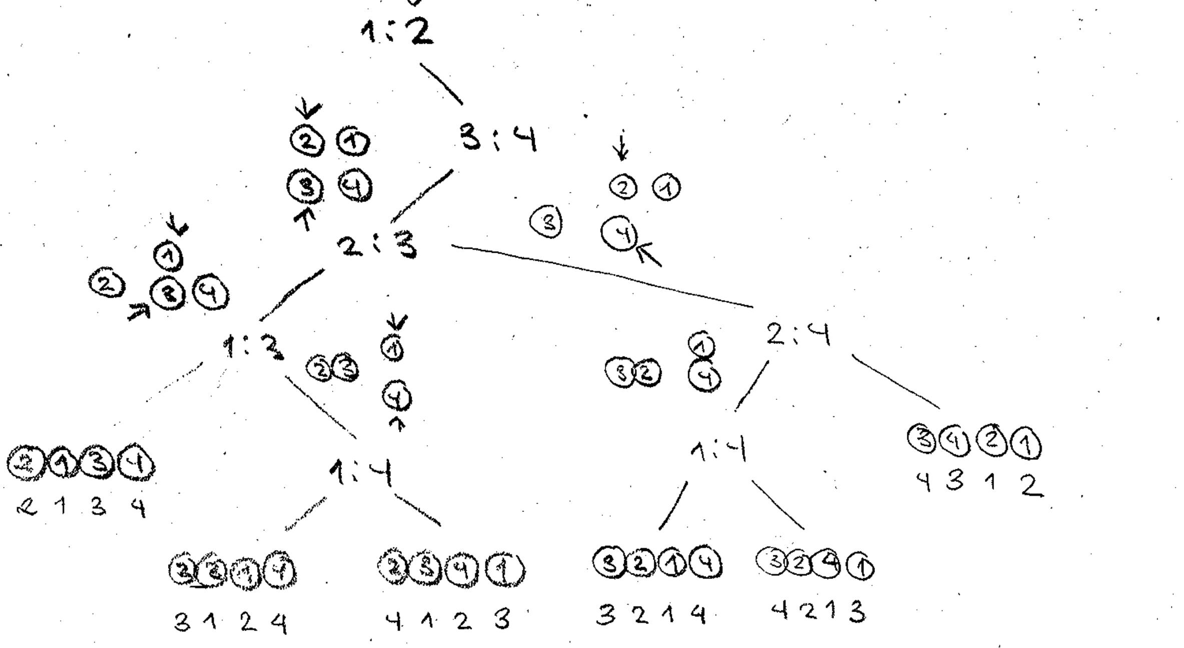
$$T(N) \leq N^{1/3} + 2\left(\left\lfloor \frac{N}{8} \right\rfloor^{1/3} - 1\right)$$ es creciente

$$\leq N^{1/3} + 2\left(\left(\frac{N}{8}\right)^{1/3} - 1\right)$$

$$= 2(N^{1/3} - 1)$$

#5.

M Sort descompone una tabla ①②③④ en ①② y ③④, ordena ambas y combina sus resultados. Por ello, tras 1:2 se efectúa 3:4 y se tiene



24

#6.a. Se tiene 6

$$A^e(N,m) = \frac{1}{N}\sum_{1}^{N} A^f(i-1,m) = \frac{1}{N}\sum_{0}^{N-1} \frac{1}{\left(1-\frac{i}{m}\right)^3}$$

$$\simeq \frac{1}{N}\int_0^N \frac{dx}{\left(1-\underbrace{\frac{x}{m}}_{t=u,\ dx=m\,du}\right)^3} = \frac{1}{N}\int_0^{N/m} \frac{m\,du}{(1-u)^3}$$

$$= \frac{1}{N/m}\int_0^{N/m} \frac{du}{(1-u)^3} = \frac{1}{\lambda}\frac{1}{2}\left[\frac{1}{(1-u)^2}\right]_0^{\lambda} = \frac{1}{2\lambda}\left[\frac{1}{(1-\lambda)^2} - 1\right]$$

$$= \frac{1}{2\lambda}\,\frac{1-1+2\lambda-\lambda^2}{(1-\lambda)^2} = \frac{2-\lambda}{2(1-\lambda)^2}$$

#6.b Si se tiene $T = [\underline{1..}\ N/2\ \underbrace{T^2}_{N/2\text{ elems}}] = [T^1, T^2]$

se tiene

$$n_{IS}(T) = n_{IS}(T^1) + n_{IS}(T^2) = N/2 + n_{IS}(T^2)$$

$$\simeq \frac{N}{2} + A_{IS}(N/2) = \frac{N}{2} + \frac{1}{4}\left(\frac{N}{2}\right)^2 + O(N)$$

$$= \frac{1}{16}N^2 + O(N)$$ 3

Análogamente, si $T^1 = [N/2 .. 1]$, se tiene

$$n_{IS}(T) = \underbrace{\frac{1}{2}\left(\frac{N}{2}\right)^2 + O(N)}_{n(T^1)} + \underbrace{n(T^2)}_{\simeq A_{IS}(N/2)}$$ 3

$$= \frac{N^2}{8} + \frac{N^2}{16} + O(N) = \frac{3}{16}N^2 + O(N)$$