Análisis de Algoritmos, Gr. Ing. Informática
Examen Final, Primera Parte, Enero 2021

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

1	2	P. 1

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 devolverá **TODAS ELLAS** al terminar el examen, separando cuidadosamente las hojas a corregir de las usadas como borrador (**ha de extremarse el cuidado en dicha separación, pues los borradores NO se corregirán en ningún caso**). 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, manteniendo los mismos fuera del alcance visual o físico de otros estudiantes. El no hacerlo así se considerará como indicio de participación en copia.
3. Las incidencias de copia detectadas durante el examen o en su corrección se pondrán en conocimiento de la Dirección de la EPS para aplicación de la normativa correspondiente,
4. Los estudiantes que **NO se hayan presentado a algún parcial** y no hayan seguido por tanto el itinerario de evaluación continua, deberán obtener una puntuación media de 7 en las preguntas 1 de cada parte del examen.
5. **Sólo se tendrán en cuenta aquellas respuestas debidamente razonadas, salvo que se indique lo contrario**

Preguntas

1. a. (2 puntos) i. Enunciar los casos peor y medio del algoritmo InsertSort.

   ii. Si una tabla σ tiene 100 elementos y 4.500 inversiones, ¿cuántas inversiones tiene σ^t? ¿Y $(\sigma^t)^t$?

   b. (3 puntos) Si $T_1 \sim f$ y $T_2 \sim f^2$, todas ellas funciones crecientes, positivas y que tienden a ∞ con N, argumentar la verdad o falsedad de las siguientes afirmaciones:
   i. $\frac{T_1^2}{T_2} \sim 1$.
   ii. $T_2 = f^2 + \mathrm{O}(T_1)$.
   iii. $T_1 = \mathrm{o}(T_2)$.

   c. (5 puntos) De una cierta tabla se sabe que la probabilidad de tener que buscar en ella el elemento i–ésimo es

   $$P(K = T[i]) = \frac{1}{C_N} i^{-\frac{3}{4}},$$

   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 $A^e_{BL}(N)$ de las búsquedas lineales realizadas con éxito sobre dicha tabla.

2. a. (3 puntos) i. Enunciar **directamente** con qué funciones son asintóticamente equivalentes las sumas $\sum_1^N \log n$ y $\sum_1^N \frac{1}{n}$?

   ii. Un algoritmo de coste n^2 tarda un segundo sobre una tabla de 1.000 elementos. ¿Cuánto tardará sobre una tabla de 10.000 elementos? ¿Y cuál debería ser el tamaño de una tabla para que tarde sobre la misma 4 segundos?

   iii. ¿Cuál es el coste en el caso peor del algoritmo BubbleSort con control de swaps sobre tablas de N elementos? ¿Y el coste en el caso mejor?

   b. (4 puntos) ¿Cuántas comparaciones de clave haría como mínimo cualquier algoritmo de ordenación local sobre la permutación $\sigma = [K, ..., 1,\ \ 2K+1, ..., 3K-1, 3K,\ \ 2K, 2K-1, ...K+1]$ de $N = 3K$ elementos? ¿Y sobre su **traspuesta** σ^t?

   c. (3 puntos) El siguiente pseudocódigo corresponde a un algoritmo que calcula simultáneamente el máximo y el mínimo de una tabla t de n elementos:

```
int max_min(int t, int n):
    v_max = v_min = t[0]
    para i de 1 a n-1:
        if t[i] > v_max:
            v_max = t[i]
        else if t[i] < v_min:
            v_min = t[i]

    return i_max, i_min
```

   Tomando la comparación de claves como operación básica, calcular razonadamente cuántas veces se aplicará la misma como máximo y como mínimo.

Análisis de Algoritmos, Gr. Ing. Informática
Examen Final, Segunda Parte, Enero 2021

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

1	2	P. 2

Preguntas

1. a. (3 puntos) i. ¿Cuál es la profundidad del árbol de decisión de QuickSort sobre tablas de 30 elementos?

   ii. Vamos a aplicar la rutina de creación de un Max Heap a una tabla con 8 elementos. Indicar cuántas comparaciones de clave se harán como máximo durante dicha creación.

   iii. ¿Cuál es la longitud de caminos externos de un árbol binario completo de profundidad 10?

   b. (3 puntos) Tras crear un max heap sobre una cierta permutación se llega a la siguiente tabla `[18 15 4 12 11 3]`.
   Argumentar que la misma es en efecto un max heap y ordenarla según la segunda parte del algoritmo HeapSort indicando adecuadamente los pasos dados.

   c. (4 puntos) Estimar razonadamente el crecimiento de una función positiva T que cumple $T(1) = 0$ y

   $$T(N) \leq \sqrt{N} + 9T\left(\left\lfloor \frac{N}{3} \right\rfloor\right).$$

   Estimar primero un posible crecimiento en un caso particular adecuado y usarlo a continuación para estimar el crecimiento en el caso general.

2. a. (2 puntos) i. ¿Qué relación hay entre el caso medio $A_{XS}(N)$ sobre tablas de N elementos de un algoritmo XS de ordenación por comparación de claves y la longitud de caminos externos *lce* de su árbol de decisión T^N_{XS}?

   ii.¿Con qué permutación se alcanzaría el caso peor $W_{QS}(7)$ del algoritmo QuickSort?

   b. (4 puntos) El siguiente pseudocódigo corresponde a una versión `msort_inv` de MergeSort donde se ordena **primero la segunda** tabla y luego la primera.

```
def msort_inv(tabla t, indice p, indice u):
    if p == u:
        return OK

    else:
        m = (p+u)/2             //division entera
        t_1 = msort_inv(t, m+1, u)
        t_2 = msort_inv(t, p, m)
        return combinar(t_1, t_2, t)
```

   Dar el subárbol de decisión para tablas de 4 elementos cuando se aplica a permutaciones σ en las que $\sigma(2) = 4$.

   c. (4 puntos) El siguiente pseudocódigo recursivo calcula el valor del n-esimo número de Fibonacci:

```
int fib(int t):
    if n == 0 or n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

   Queremos estimar razonadamente en función de n cuántas **sumas** efectuará dicho algoritmo para calcular el n-ésimo número de Fibonacci. Para ello, dar en primer lugar dichos números de sumas para $n = 0, 1, 2, 3$ y 4 y estimar a continuación el valor del número general de sumas.

Análisis de Algoritmos, Gr. Ing. Informática
Examen Final, Tercera Parte, Enero 2021

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

P. 1	P. 2	1	2	P. 3	T

Preguntas

1. a. (3 puntos) i. ¿Cuáles son los casos peor y medio del algoritmo de búsqueda binaria sobre tablas de N elementos?

   ii. Si bien el coste de la búsqueda binaria sobre tablas ordenadas es óptimo, no es una buena idea usar dichas tablas como estructura de datos para el TAD Diccionario. ¿Por qué?

   iii. Indicar brevemente los pasos a dar para borrar de un árbol binario de búsqueda un nodo con dos hijos.

   b. (4 puntos) Construye razonadamente el árbol AVL asociado a la lista
   `[20 21 13 10 11 6 9]`
   indicando en cada paso con detalle suficiente los desequilibrios a arreglar y las acciones a tomar.

   c. (3 puntos) Queremos construir una tabla hash con encadenamiento para almacenar un cierto número de datos N utilizando una tabla de 300 punteros. ¿Cuál es el valor máximo de N para que el número medio de sondeos en búsquedas con y sin éxito sea a lo sumo 1.5? ¿Y cuál sería ese valor máximo si se tratara de una tabla hash con direccionamiento abierto y sondeos aleatorios?

2. a. (3 puntos) i. ¿Cuál es el orden del la profundidad de un AVL con N nodos?

   ii. ¿Cómo definirías una función hash de **multiplicación** que para un entero M proporcione valores entre 0 y $M-1$?

   iii. Para una cierto algoritmo hash con encadenamiento H se cumple que $A_H^f(N,m) = \phi(\lambda)$ para una cierta función ϕ y $\lambda = N/m$, donde N es el número de datos a almacenar y m es el número de punteros. ¿Cómo podríamos expresar $A_H^e(N,m)$ en términos de ϕ?

   b. (3 puntos) Vamos a usar una tabla de 11 elementos, la función hash $h(k) = k\,\%11$, índices entre 0 y 10 y encadenamiento para almacenar las claves `[17, 11, 22, 6, 3, 14]`. Indicar el estado final de la tabla en cuestión así como el número de colisiones que han tenido lugar durante su construcción.

   c. (4 puntos) De un cierto algoritmo hash H con **direccionamiento abierto** se sabe que

   $$A_H^f(N,m) = \frac{1}{(1-\lambda)^2},$$

   con $\lambda = N/m$ el factor de carga.

   Deducir razonadamente cuál será la fórmula para el número medio $A_H^e(N,m)$ de sondeos necesarios para buscar con éxito en una tabla de N datos y m posiciones. ¿Cuál será el número medio de sondeos en búsquedas con éxito para tablas con un factor de carga $\lambda = 0{,}5$?