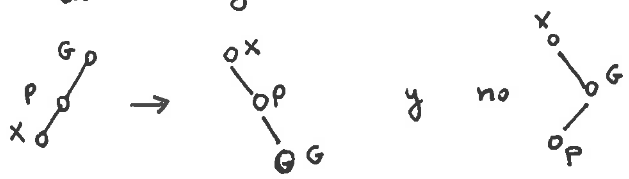
ARBOLES DE ENSANCHAMIENTO (SPLAY TREES) Weiss Ch. 21

- Arboles con ensanchamiento ascendente

- Caso simple: hijo de la raíz
  - Lo rotamos hacia la raíz (zig)
- Caso zig-zag: hijo y nieto a lados diferentes
  - Una rotación doble lo lleva a la raíz (zig-zag)
- Caso zig-zig: hijo y nieto por el mismo lado
  - En vez de realizar una rotación doble ascendente la realizamos (localmente) en modo descendente



Operaciones

- Inserción seguida de ensanchamiento, que conduce a la raíz al elemento insertado.
- Búsqueda seguida de ensanchamiento, para llevar a la raíz al elemento encontrado o a su (fallido) padre.
- Eliminación
  - Búsqueda que lleva a la raíz el elemento a eliminar
  - Partimos el árbol en dos
  - Buscamos (¡ensanchando!) el mayor del primero
  - Le pegamos por la derecha el segundo.

- Arboles con ensanchamiento descendente

- Descensos que van manejando tres subárboles (L, X, R)

  Nos movemos en X y mantenemos el invariante de que éste junto a L (que sólo contiene elementos menores que los de X) y a R (resp. mayores) es igual al árbol original.

  Iniciamos pues con (vacío, A, vacío).

- Buscamos el elemento en X bajando uno o dos niveles y encontrado el mismo, o en la ruta hacia él, hacemos:

  - zig descendente al terminar en Y hijo de X, mandando el hijo descabalgado al correspondiente remanente L o R.
  - zig-zag descendente: se puede simplemente aplicar el zig.
  - zig-zig descendente: mantenemos el subárbol que pende del nieto como central; rotamos el resto mandándoselo al correspondiente remanente.

- Pegamos adecuadamente el central y los remanentes manteniendo la raíz del central y colgando sus hijos de las hojas extendidas correspondientes al máximo (resp. mínimo) de L (resp. R).

Coste amortizado del ensanchamiento $3 \log N + 1$

$S(i)$ = tamaño del subárbol con raíz en i

$\Phi(T) = \sum \log S(i)$; $R(i) = \log(S(i))$; $\Phi(T_0) = 0$

Tma. 21.3 : Una inserción agrega como mucho $\log N$ al potencial.

Tma 21.2 : $\Phi_i - \Phi_{i+1} + r_i \le 3 \log N + 1 \Rightarrow \sum_1^M r_i \le (3 \log N + 1) M$
para ensanchamientos.

Una inserción hace que Φ crezca menos de $\log N$

$\Phi_{i(ins)} - (\Phi_{i-1} + \log N) \le 3 \log N + 1$ $\quad \Phi_{i(ins)} - \Phi_{i-1} \le 4 \log N + 1$.

$\sum_1^M r_i$ (busquedas e inserciones) $\le (3 \log N + 1) M_b + (4 \log N + 1) M_i$

Nota: Cada bajada en una inserción o busqueda cuesta "lo que" sus rotaciones.

Tma 21.4 : $a + b \le c \Rightarrow \log a + \log b \le \log c - 2$

$\sqrt{ab} \le \frac{c}{2}$ y tomamos logaritmos.

Tma 21.5 : Paso zig $\Delta_\Phi \le 3(R_f(X) - R_i(x))$

Demost : Solo cambian X y su padre P. $R(X)$ crece y $R(P)$ decrece.

Tma 21.6 : Paso zig-zag $\Delta \Phi \le 3(R_f(X) - R_i(x)) - 2$

Demost : Cambian X, P y G. Manipulaciones sencillas de Δ usando Tm. 21.5.

Tma 21.7 : Paso zig-zig $\Delta \Phi \le 3(R_f(X) - R_i(X)) - 2$ Dem. similar.

Demost (Tma 21.2) $\quad \Delta_\Phi^{ens} \le 3(R_f^{ens}(X) - R_i^{ens}(X)) - 2k$

Rotaciones totales $2k + 100$ Beautiful !!

$\Delta_\Phi^{ens} + (2k + 100) \le 3 \log N + 1$

Las rotaciones dobles son las que amortiguan el crecimiento de Φ !!

Nota: Las rotaciones dobles son las que amortiguan el crecimiento de Φ !!