Universitat de Lleida
Grau en Tècniques d'Interacció Digital i de Computació
Estructura de Dades

Pràctica 5

Heap (monticle): una cua de prioritat

Cristian Oprea i Xavier Vila

10 de desembre de 2020

1 L'estratègia

L'estratègia emprada per a resoldre aquesta pràctica és molt senzilla. Partíem de la idea de que si enumerem els nodes d'un arbre binari equilibrat i complert, si el pare és n, el fill esquerra serà $2n$ i el fill dret $2n + 1$. Aleshores ja teniem tot el necessari per a convertir un arbre binari en un array i poder-lo treballar còmodament.

2 Mètode add

Per a afegir un element al monticle primer és necessari crear el forat on anirà aquest element, que al treballar amb una array això es tradueix en augmentar en u la seva capacitat, en el nostre cas la seva mida actual (`currentSize`). Després l'afegim al format que hi hem fet, l'última posició de l'array, i mentre el seu pare sigui més gran que ell l'anem pujant (utilitzant la numeració explicada a L'estratègia), ja que si no no seria un monticle.

3 Mètode remove

Per eliminar un element del monticle s'empra una estratègia semblant a la del Mètode add, però a l'invers. Primer de tot eliminem l'arrel i la "substituïm" amb l'últim element que haguem afegit. Anirem mirant, des de l'arrel, quin dels dos fills és més petit i si aquest també és més petit que el node que haguem triat (en el nostre cas, l'últim afegit) el pujarem, desplaçant així el forat que ens ha quedat a l'arrel. Fins quan? Fins que els dos fills siguin més grans que el node triat o fins que arribem a l'últim nivell de l'arbre.