

SI-möte #11, Programkonstruktion och datastrukturer

Elias Castegren

elca7381@student.uu.se

15 februari 2011

Begrepp

- i)* Vilka egenskaper har en *heap*? Vad är det karaktäristiska för elementens placering i en min-heap (eng. *heap property*)?
- ii)* Vad är en *binär heap*? Kan en binär heap vara tom? Hur många element (som mest) innehåller en binär heap av höjd h ? Hur många element finns på nivå i i samma heap?
- iii)* Vad är ett *binomialt träd*? Vad menas med rangen av ett binomialt träd? Hur många noder har ett binomialt träd av rang k ? Hur många noder har trädet på nivå i och vad har trädet för höjd? Kan ett binomialt träd vara tomt?
- iv)* Vad är en *binomial heap*? Hur är träden i en binomial heap ordnade och vad gäller för trädens rang? Kan en binomial heap vara tom?
- v)* Är en binomial heap ett träd?

Övningar

1.

Lägg in nedanstående element (i given ordning) i en binomial min-heap.

4, 3, 1, 7, 2, 6, 5

Skriv upp hur heapen ser ut i varje steg (det kan underlätta att rita upp heapen även i mellanstegen när träden struktureras om).

2.

Ta bort det minsta elementet ur den resulterade heapen i uppgift 1 och se till att heapstrukturen behålls.

3.

Gör samma saker som i uppgift 1 och 2 men med en binär max-heap istället (Ta bort det största elementet istället för det minsta).

4.

Ange rangerna hos de binomiala träden i en binomial heap med

i) 52 element

ii) 53 element

iii) 127 element

iv) 128 element

Hur ser det värsta fallet ut när man ska lägga till ett element i en binomial heap och kring vilka värden ligger antalet element i heapen då?

5.

Beskriv hur man kan använda en heap (binär eller binomial) som “mellanlandning” för elementen när man sorterar en lista. Ge en informell härledning av tidskomplexiteten i det värsta fallet. (Algoritmen kallas *heapsort*)

6.

Implementera binomiala heapar i ML-kod. Skriv funktionerna `insert(t, h)` som sätter in det binomiala trädet `t` i heapen `h`, `extractMin(h)` som returnerar och tar bort det minsta elementet i `h` och `search(k, h)` som returnerar (utan att ta bort) noden med nyckel `k` taggad med `SOME` ur `h`, annars `NONE`. Utgå från nedanstående definitioner:

```
datatype 'a binoTree = Node of int * int * 'a * 'a binoTree list
type 'a binoHeap = 'a binoTree list
```

I det binomiala trädet `Node(r, k, d, l)` är `r` trädets rang, `k` rotnodens nyckel, `d` datan i rotnoden och `l` listan av underträd (med fallande rang $r - 1, r - 2, \dots, 1, 0$). Tänk på att bevara och följa (och utnyttja) invarianter och förvillkor.

Lycka till!