

Tema 5: prioritetsköer

Elise Edette `tero0337`

Emma Persson `empe5691`

Aframyeos Rohoum `afro0793`

12 februari 2015

1 Ändringar som krävdes för att omvandla en BinaryHeap till en D-heap

Konstruktorn ändrades så att den tar emot en *int* variabel som säger hur många barn en nod skall ha. Den gamla konstruktorn tog emot en *int* för array storleken, vi använder en intern konstant för det istället.

I *insert()* metoden ändrades uträkningen av föräldernoden (under percolate up). Istället för formeln

$$1 \parallel i / 2$$

så använder vi

$$1 \parallel (i + (d - 2)) / d$$

där d är antalet barn. I metoden *percolateDown()* ändrades det så att istället för att ha en if-sats för att välja en av de två barnen så har vi en loop som söker igenom alla barn för att hitta den som har högst prioritet.

För att hitta det första barnet ändrades formeln från

```
1 || i * 2
```

till

```
1 || d * (i - 1) + 2
```

Utöver det lades det till små metoder som *size()* och *get()*.

2 Muntafrågor

1. Vad är skillnaden mellan en binärheap och en dheap? VG:
Beskriv skillnaden på kostnaden (ordo komplexitet) för att lägga till och ta bort saker i en binärheap vs dheap.
2. Beskriv begreppen percolate up och percolate down? VG:
beskriv, steg för steg, vad som händer i percolate down.