Advanced computer programming

Exercise session 4: Heaps, Priority queues and trees

Jean-Michel Begon

November 2014

Exercice 1

(a) How many elements are there at the least (resp. at the most) in a heap of height n ?

(b) Where is situated the smallest element in a max-heap ?

(c) What is the relationship between a min-heap and a sorted array ?

(d) Is this array a max-heap : $[23, 17, 14, 6, 13, 10, 1, 5, 7, 12]$?

Exercise 2

Draw all the min-heaps for the set of keys $\{A, B, C, D, E\}$.

Exercise 3

Let `H` be an initially empty min-heap. Draw the evolution of heap's state with respect to the following operations :

```
heapInsert(H, 5)
heapInsert(H, 4)
heapInsert(H, 7)
heapInsert(H, 1)
heapExtractMin(H)
heapInsert(H, 3)
heapInsert(H, 6)
heapExtractMin(H)
heapExtractMin(H)
heapInsert(H, 8)
heapExtractMin(H)
heapInsert(H, 2)
heapExtractMin(H)
heapExtractMin(H)
```

Exercise 4

(a) How to implement a stack thanks to a priority queue ?

(b) How to implement a queue thanks to a priority queue ?

(c) How to implement a random queue thanks to a priority queue ?

(d) How to implement a maximum priority queue thanks to a min-heap ?

(e) How to implement a priority queue with a complexity of $\Theta(1)$ for insertions and $O(n)$ for extractions ?

Exercise 5

The preorder walk of a given binary search tree `T` yields :

```
A B C - - D - - E - F - -
```

The letters correspond to internal nodes and dashes represent leaves. What would yield the inorder and postorder tree walks ?

Exercise 6

Implement the depth-first and breadth-first tree walks for heaps.

Exercise 7

Depth-first traversal relies on the stack while breadth-first uses a queue. What would happen if we were to use a maximum priority queue instead ?

Exercise 8

Let `T` be a tree.

(a) Write a function `size(T)` which determines the number of element in the tree.

(b) Write a function `isComplete(T)` which determines whether the tree is complete or not.

(c) Write a function `mirror(T)` which returns the mirror tree of `T`.

(d) Write a function `printLowerThan(T, x)` which prints the values in `T` lower or equal to x.

What is the complexity of these operations ?

Bonus : For all of these functions, write an algorithm which is a) recursive et b) iterative. *Bonus :* Why are the recursive implementation so natural ?

Exercise 9

Prove that the number of leaves in a binary tree is equal to the number of nodes of degree 2, plus 1.

Bonus

Bonus 1

Priority queues implemented with heaps offer interesting performances but have a bounded capacity. We would like to craft an extensible heap to overcome this limitation.

(a) Propose a pseudo-code for this task.

(b) How have the performances evolved ?

Bonus 2

A *d-ary* heap is similar to a binary heap except that each internal node has d children.

(a) Extend the array representation for the *d-ary* heap.

(b) Draw the ternary heap corresponding to the following array :
$[25, 12, 24, 15, 8, 7, 4, 13, 6, 3, 5, 9, 10, 1, 2]$.
Is it a max-heap ?

(c) What is the height of a *d-ary* heap of n elements ? (With respect to d and n)