Lecture 4 – Priority Queues

Reading: KT Section 2.5

Priority queues, which data structure should we use?

• It’d be great if we can keep the elements sorted.

• Array?

• Linked list?

• Binary Tree? Binary Search Tree?

But then... what kind of tree is this?

• A minheap is a
  • complete binary tree
  • each element is less than or equal to both of its children

• A minheap keeps the smallest valued element readily available
Finding child nodes


Finding parent nodes

Parent of H[10] is at H[5]

Adding a new element

Heapify-up(H, 15)
Algorithm 2.8, page 61

Heapify-up(H, i):
If i > 1 then
  let / = parent(i) = \lfloor i/2 \rfloor
  If key[H[i]] < key[H[j]] then
    swap the array entries H[i] and H[j]
    Heapify-up(H, j)
  Endif
Endif

Why does this work?
Deleting an element

Heapify-down(H, 3)

Heapify-down(H, 7)

Algorithm 2.9, page 63

Why does this work?