

Binary Heaps & Priority Queues

CPSCP, 12.05.26

Non-Linear Data Structures

- In linear data structures, e.g. linked lists / arrays, operations typically take constant or linear time

	Array	Linked list
Access(i)	$O(1)$	$O(n)$
Search(x)	$O(n)$	$O(n)$
Insert(x)	$O(1)$	$O(1)$
Delete(x)	$O(n)$	$O(n)$

- In non-linear data structures we usually get some operations done in $O(\log n)$

- This

[90, 19, 36, 17, 3, 25, 1, 2, 7]

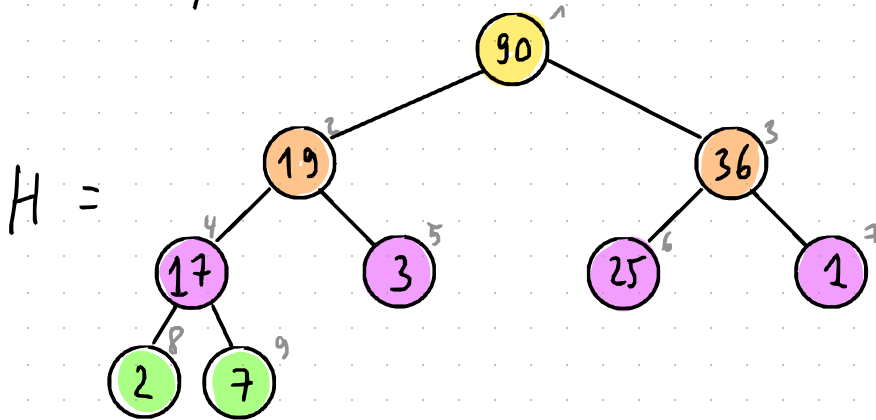
is a Binary heap

- Looks pretty linear and not particularly binary, but ...

Binary Heaps

• $H = [90, 19, 36, 17, 3, 25, 1, 2, 7]$

actually "means"



- numbers in array are the nodes of a complete binary tree level-wise
↑ up to last level

• How to "navigate" tree?

- left(i): $2i$ $i << 1$
- right(i): $2i+1$ $(i << 1) + 1$
- parent(i): $\lfloor i/2 \rfloor$ $i >> 1$

• What is the height of H?

- $O(\log n)$

• Max Heap Property

For every node x , all nodes in the left and right subtree of x are at most x .

- Min Heap: same but "at least"
- Where is the max. element in a Max Heap?

Operations on Binary Heaps

• Extract - Max: $O(\log n)$

- Delete element in root
- Put last leaf in its place
- Iteratively swap element with larger child downwards

• Insert: $O(\log n)$

- Insert element as last leaf
- Iteratively swap element with parent upwards

• Two main use cases

1. Sorting

2. Priority Queues

return the
k largest elements
in sorted order

1. Heapsort

- build heap

• sift down on all $O(n)$ non leaf nodes

• trivially $O(n \log n)$, but actually $O(n)$

- n times Extract - Max

• $O(n \log n)$

"sift down"

First Task for Today:

Give an $O(n \log k)$ algorithm for the partial sorting problem with parameter k .

2. Priority Queues using Heaps

- can implement priority queues using binary heaps.
- typical operations:
 - `push(x)`: inserts an element
 - `pop()`: returns max or min element
 - `top()`: just gives value of top
- `push, pop` in $O(\log n)$
- `top` in $O(1)$
- In practice
 - C++: `STL priority_queue`
 - Java: `Priority Queue`
 - Python: `heapq`
- sometimes we want to update keys of elements already in the priority queue
 - need to maintain info on where in the heap the elements are

Second Task for Today:

Boiling Vegetables:

<https://open.kattis.com/problems/vegetables>