Heaps/Priority Queues

Definition: An *abstract data type* to efficiently support finding the item with the highest priority across a series of operations. The basic operations are: insert, find-minimum (or maximum), and delete-minimum (or maximum). Some implementations also efficiently support join two priority queues (*meld*), delete an arbitrary item, and increase the priority of a item (decrease-key).

Guaranteed *O(log(n))* extraction/insertion.

Binary Heaps: Based on complete binary trees (*not search trees!*).

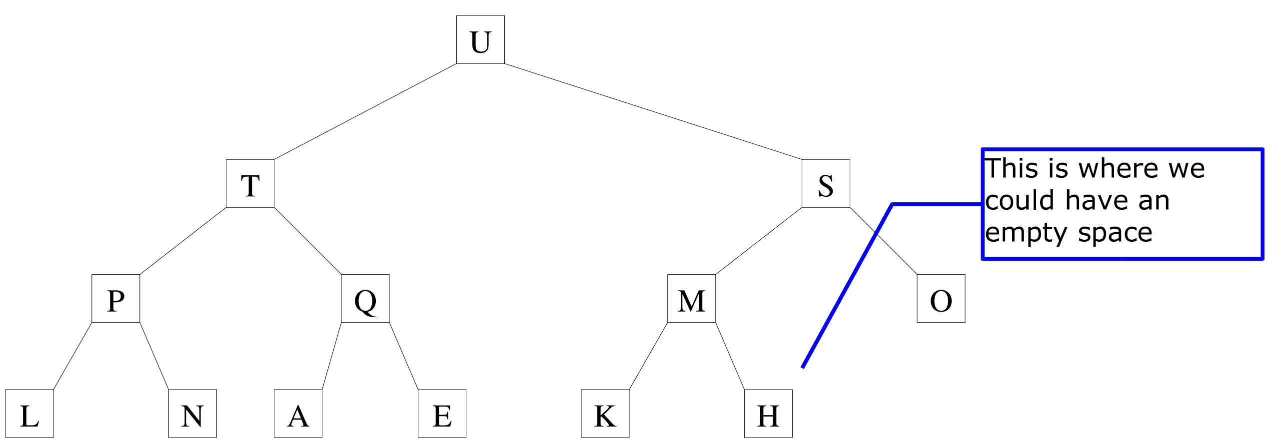
A Binary Tree has the *heap property* if
1. it is empty *or*
2. the key in the root is larger than that in either child and both subtrees have the heap property.

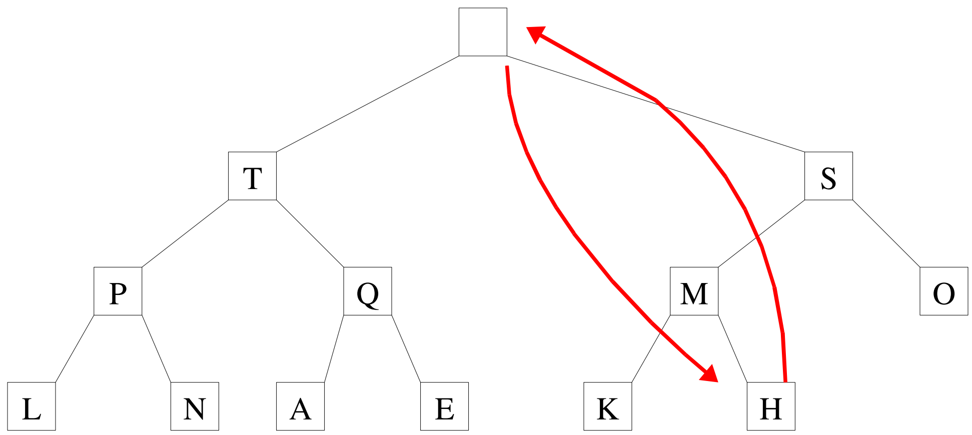
A Complete Binary Tree(CBT) is filled all the way, except possibly the right-most leaves.
A CBT of height h will have n nodes where $2^h \leq n \leq 2^{h+1} - 1$. So the height of a complete binary tree is $\lfloor \log(n) \rfloor$, which is $O(\log(n))$.

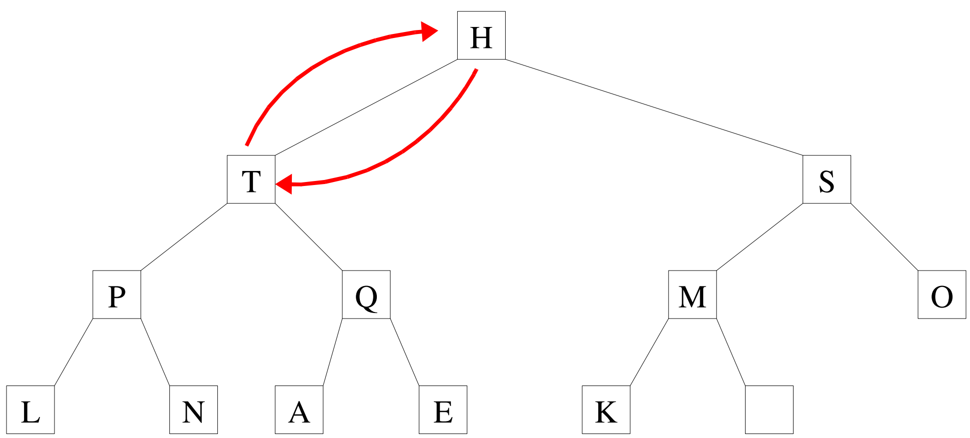
Example Uses

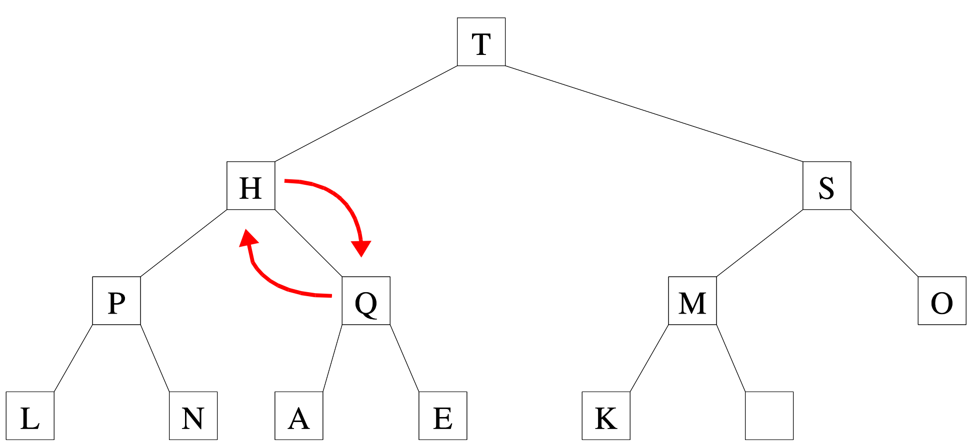
- Simulations (events are ordered by the time at which they should be executed)
- Job scheduling in computer systems (higher priority jobs should be executed first)
- Constraint systems (higher priority constraints should be satisfied before lower priority constraints)

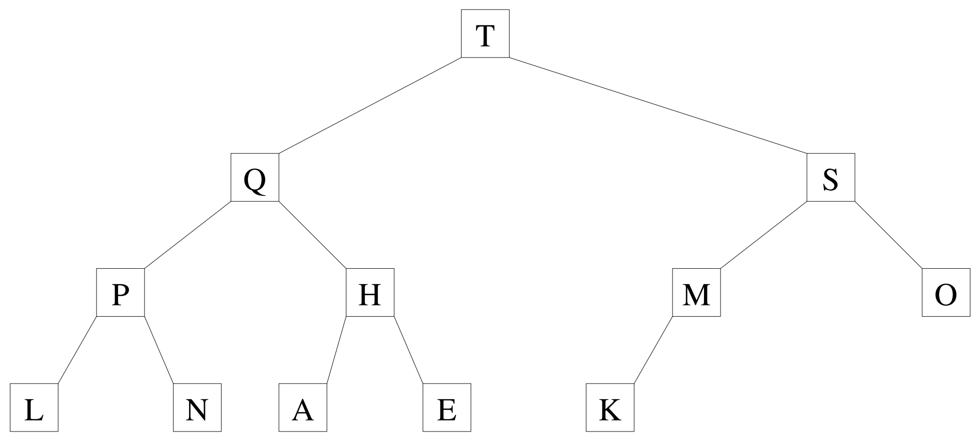
Example : Consider deletion from the Binary Heap (alphabetical priority):




T
S
P
Q
M
O
L
N
A
E
K
H

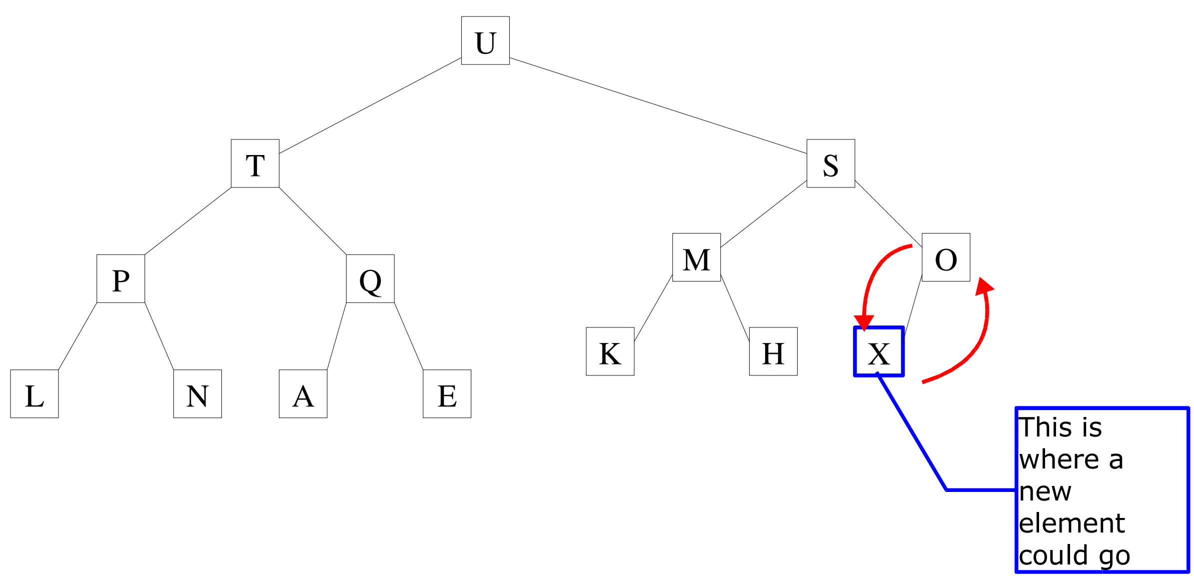

H
T
S
P
Q
M
O
L
N
A
E
K


T
H
S
P
Q
M
O
L
N
A
E
K

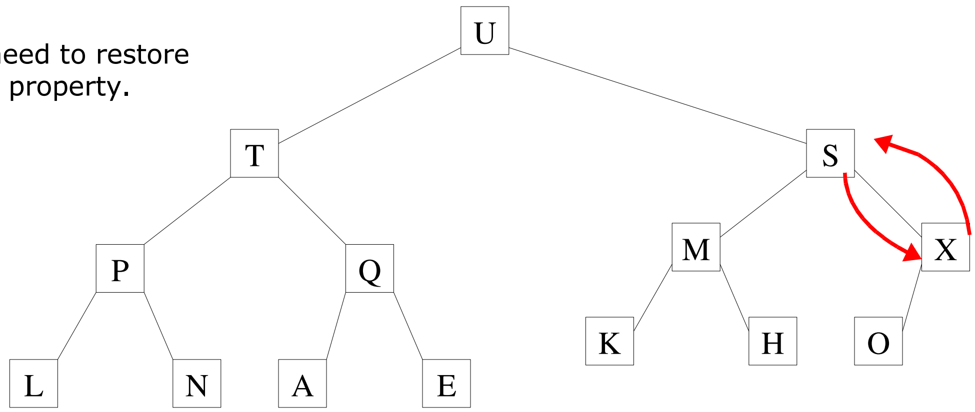


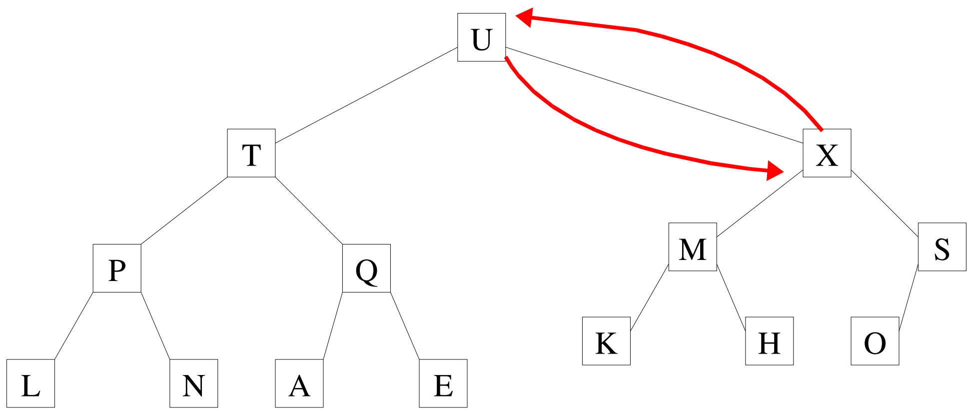
Note that we need at most h = O(log(n)) exchange operations. The deleted hole "percolates down".

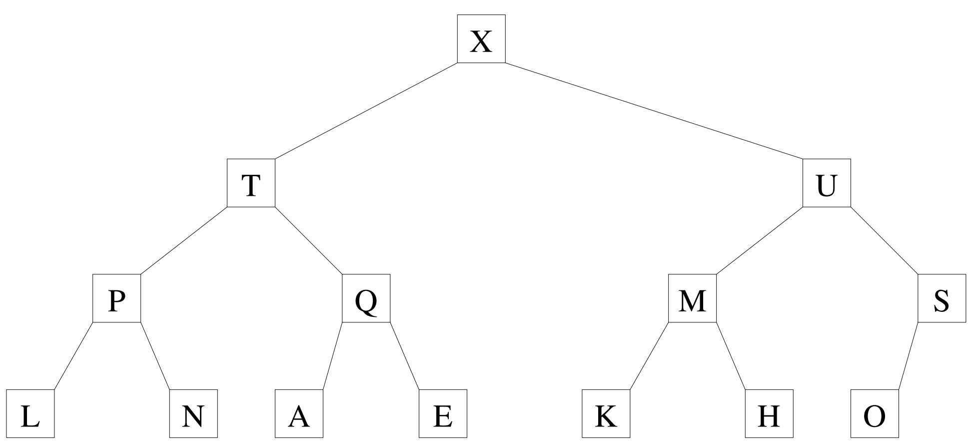
Consider inserting X into the heap:



We need to restore heap property.







So, the added vertex "percolated up".

HeapSort algorithm

To sort *n* objects, add them in a heap, one item at a time, and then remove one item at a time from the heap. The removed items will be sorted.