【DS】Day13

Tags	
Date	@June 6, 2022
Summary	Binary Heap

【Week 4】Priority Queue

4.2 Binary Heap

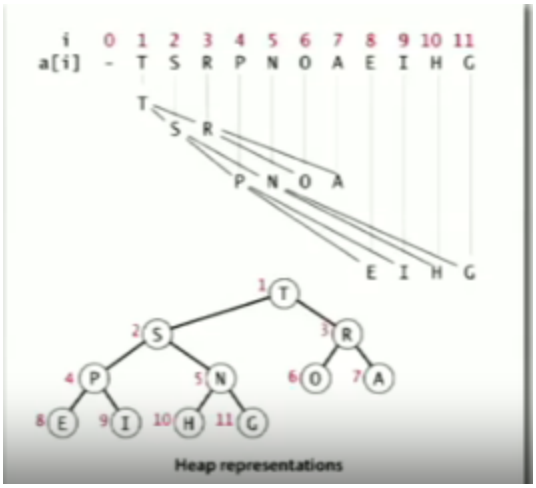
Binary Heap: Array representation of a heap-ordered complete binary tree.

Heap-ordered binary tree:

- Keys in nodes
- Parent's key no smaller than children's keys

Array representation:

- Indices start at 1
- Take nodes in level order
- No explicit links needed

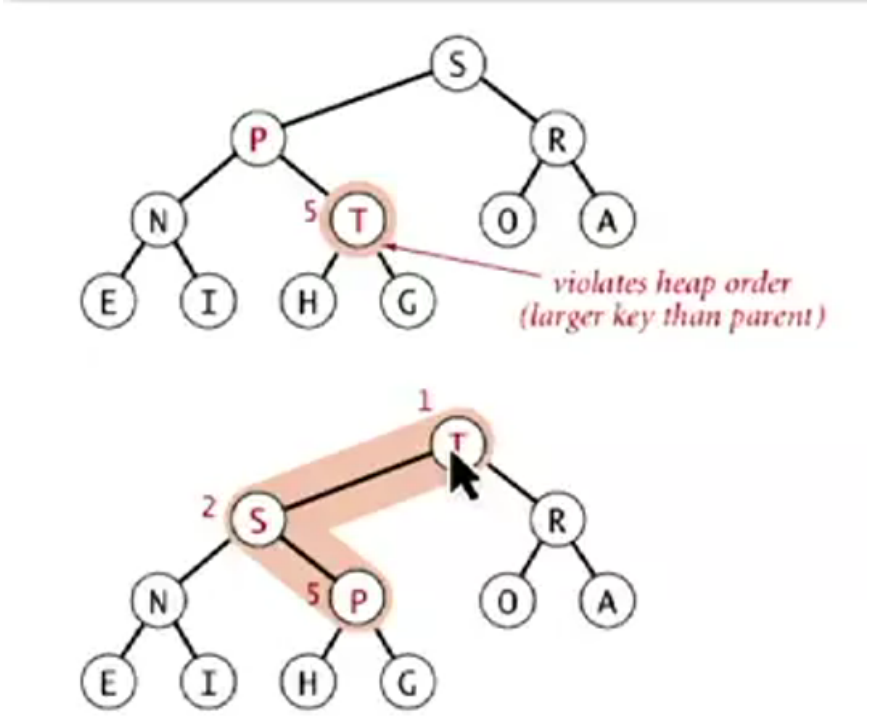


Proposition:

- Largest key is `a[1]`, which is the root of binary tree.
- Parent of node at k is at k/2.
- Children of node at k are at 2k and 2k + 1.

Promotion in a Heap

Scenario: Child's key becomes larger key than its parent's key.



To eliminate the violation:

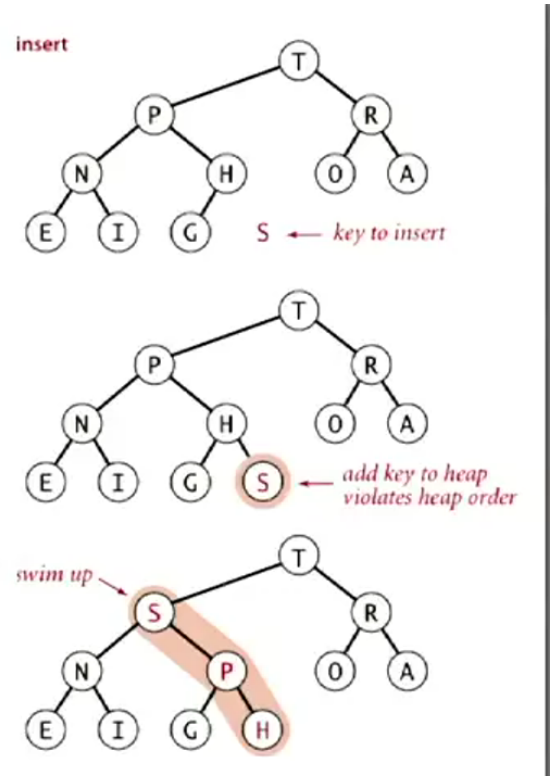
- Exchange key in child with key in parent
- Repeat until heap order restored.

```
private void swim(int k) {
  // Not at the root and k's parent's key is less than k's key
  while (k > 1 && less(k / 2, k)) {
    exch(k, k / 2);
    k /= 2;
```

```
  }
}
```

Insert: Add node at end, then swim it up

Cost: At most `1 + lgN` compares.




```
public void insert(Key x) {
  pq[++N] = x;
  swim(N);
}
```

Scenario: Parent's key becomes smaller than one(or both) of its children's.

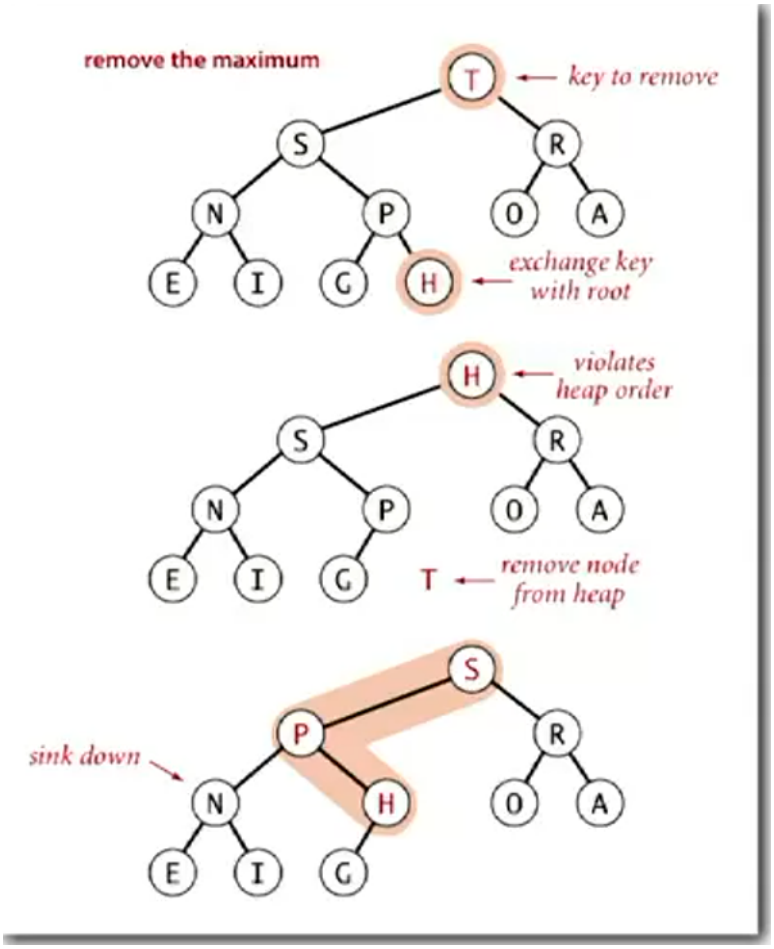
To eliminate the violation:

- Exchange key in parent with key in larger child.
- Repeat until heap order restored.

```
private void sink(int k) {
  while (2 * k <= N) {
    int j = 2 * k;
    // Select the key of the larger children's node
    if (j < N && less(j, j + 1)) ++j;
    if (!less(k, j)) break
    exch(k, j);
    k = j;
  }
}
```

Delete max: Exchange root with nodes at end, then sink it down.

Cost: At most `2lgN` compares.




```
public Key delMax() {
  Key max = pq[1];
  exch(1, N--);
```

```
  sink(1);
  pq[N + 1] = null;
  return max;
}
```