

CSE 3010 – Data Structures & Algorithms

Lecture #46

What will be covered today

- Insertion in heap
- Deletion from heap

Insertion in heap

- Uses 'trickle up'
- Place the node to be inserted in the first open position in the heap
- Trickle up the newly inserted node until it is:
 - Below a node with a larger key
 - Above a node with a smaller key

Deletion from heap

- Uses 'trickle down'
- Remove the root
- Move the last node to the root
- Trickle down the last node down until it is:
 - Below a node with a larger key
 - Above a node with a smaller key

Implementation of HEAP data structure

```
void insert(ITEM binaryHeap[], ITEM key) {
    int indexToInsert, i, size, parent, temp;
    size = sizeHeap(binaryHeap);
    indexToInsert = 0;
    for (i = 0; i <= size; i++) // Finding the location to insert the
        key
        if (binaryHeap[i] == 0)
            indexToInsert = i;
    binaryHeap[indexToInsert] = key;
    while (indexToInsert > 0) {
        if (indexToInsert % 2 == 0)
            parent = (indexToInsert - 2) / 2;
        else
            parent = (indexToInsert - 1) / 2;
        if (binaryHeap[indexToInsert] > binaryHeap[parent]) {
            temp = binaryHeap[indexToInsert];
            binaryHeap[indexToInsert] = binaryHeap[parent];
            binaryHeap[parent] = temp;
        }
        indexToInsert = parent;
    }
}
```

3/12/20