Priority Queue

Week 9

03.05.2021

1. - Linked List
   - Multi-dimensional Array
2. - Heap Tree
3. - Binary Search Tree

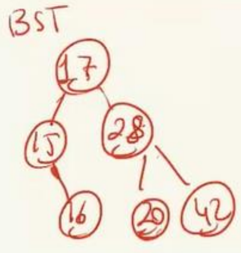
Priority Queue Applications

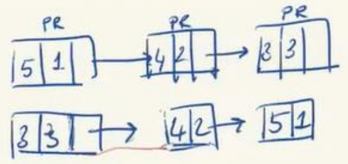
- Process Management and Interrupt Handling
- Dijkstra Algorithm (Shortest Path Algorithm)
- Data Compression (Huffmann Coding)
- Queue on banking operations
- Airplane Queues (Business Class, Economy Class)

Priority Queue Operations

- Insert
- Delete
- Peek

BST





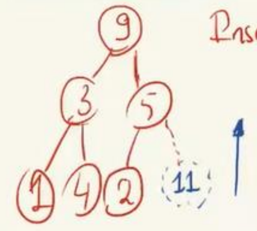
Operations	peek	insert	delete
Linked List	O(1)	O(n)	O(1)
Binary Heap	O(1)	O(logN)	O(logN)
BST	O(1)	O(logN)	O(logN)

Binary Heap

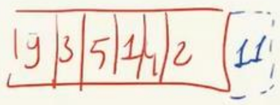
Peek

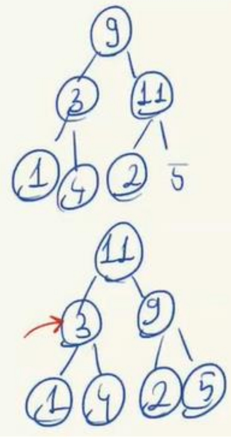
No delete

Insert



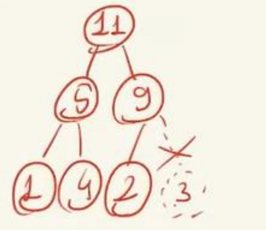
Insert 11





Delete

Remove 3

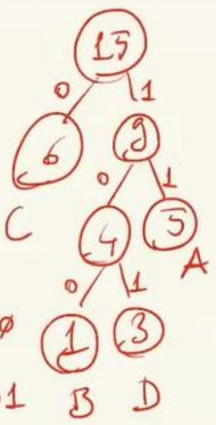


```
void insert (int array [], int newNum){
    if (size ==ø){
      array [ø]= newNum;
      size +=1;
    }
    else {
      array[size] = newNum;
      size+=1;
      for (i=size/2 -1; i>=ø; i--{
         heapify (array, size, i);
      }
    }
}
```

```
void heapify (int array[], int size, int i){
  if (size== 1){
    printf ("Single element in the heap");
  }
  else {
    largest=i;
     l= 2*i+1
     r = 2*i +2;
    if (l< size && array[l]>array[largest])
      largest=l;
    if(r<size  && array [r]>array
                               [largest])
      largest=r;
    if (largest!=i)
      swap (&array[i], &array[largest])
  }  heapify (array, size, largest);
}  }
```

```
void deleteRoot (int array[], int size){
    swap (&array[0], &array[size-1]);
    size -= 1;
    for (i=size/2-1; i>=0; i--)
        heapify (array, size, i);

}
```



C 0, A 11, B 100

D 101

Huffmann Coding

BCAADDDCCACACAC

B	C	A	D
1	6	5	3

B	D	A	C
1	3	5	6

1 3 4 5 6

1 3 4 5 6 9

1 3 4 5 6 9 15