Priority Queue

An Introduction

Md Awsaf Alam[1]
Ahmed Nafis Fuad[1]

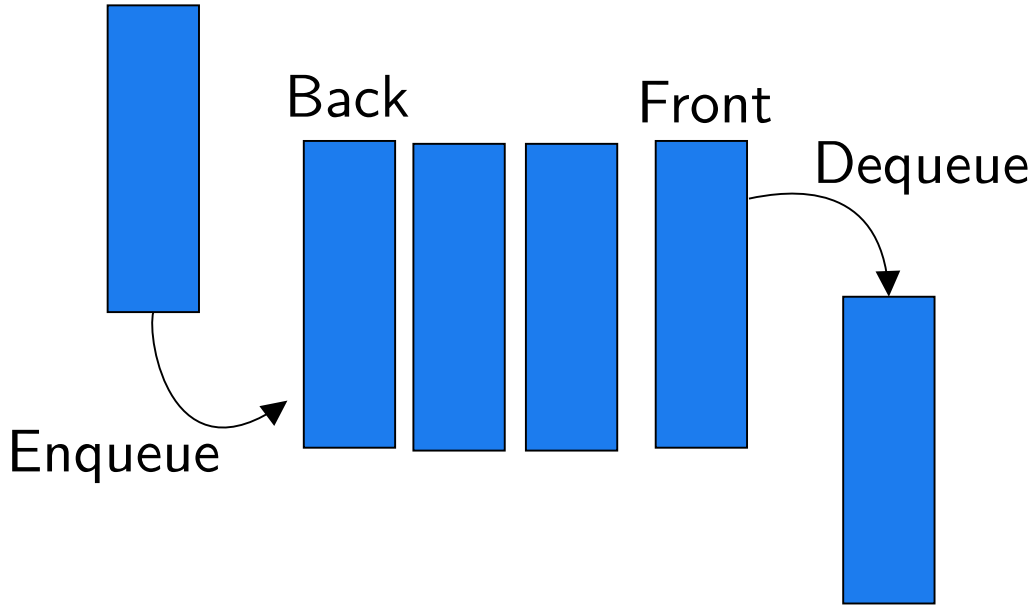
[1]Department of Computer Science and Engineering
BUET

July 23, 2018

Table of Contents

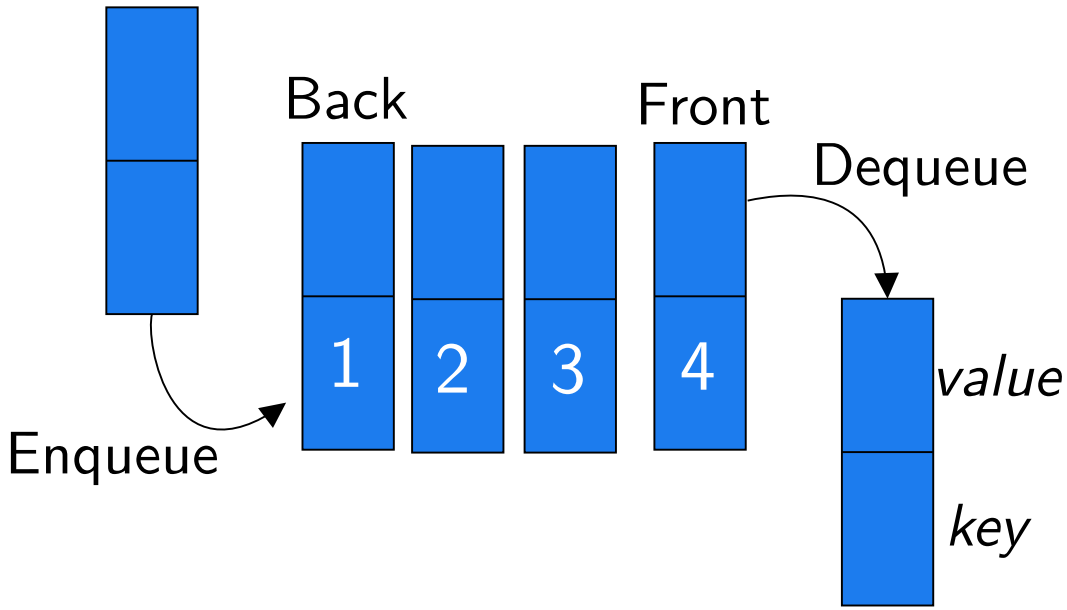
What is a Queue?

A queue is an example of a linear data structure, which works on the basis of "**first-in-first-out**" (FIFO).



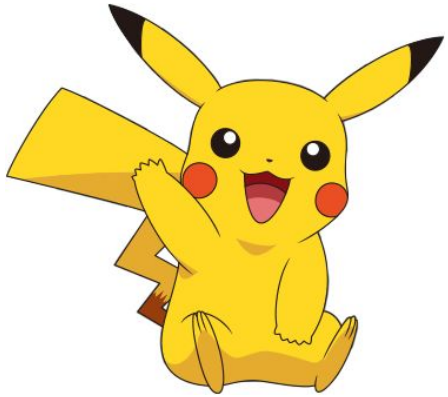
What is a Priority Queue?

A priority queue is like a regular queue or stack data structure but where additionally each element has a "**priority**" associated with it. In a priority queue, an element with high priority is served before an element with low priority.



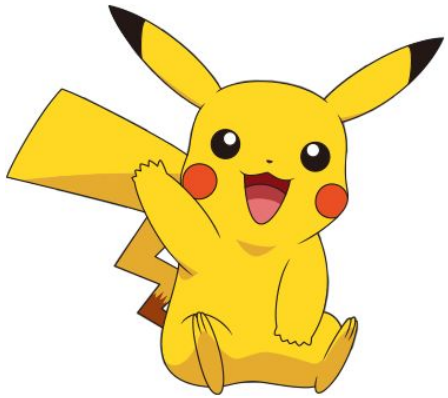
Application of Queue

Application of Queue



1

Application of Queue

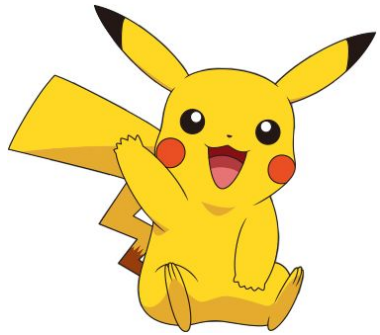


1

1

Application of Queue

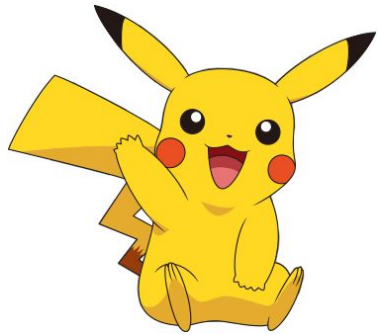
2



1

Application of Queue

2



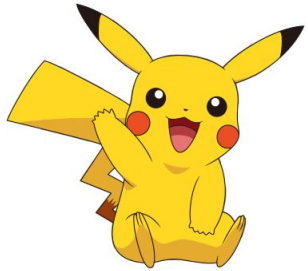
1

1

Application of Queue

3

2

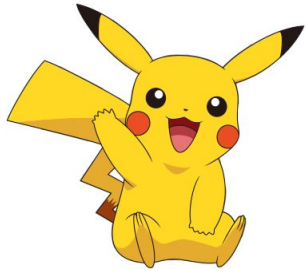


1

Application of Queue

3

2



1

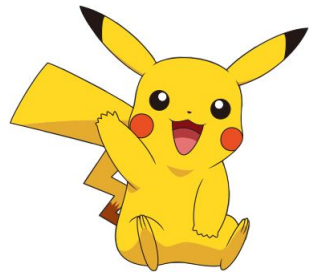
1

Application of Queue

3

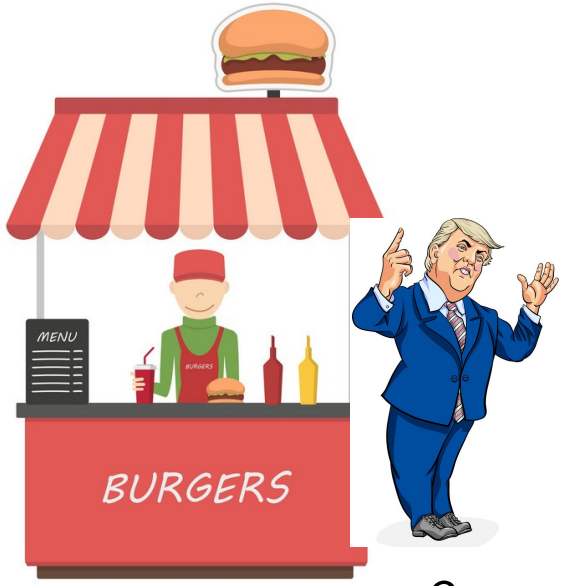
3

2



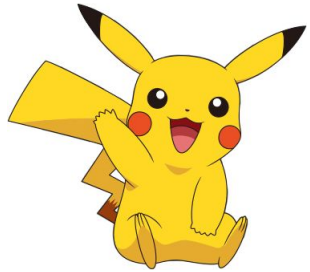
1

Application of Queue



3

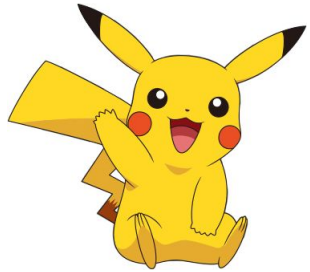
2



1

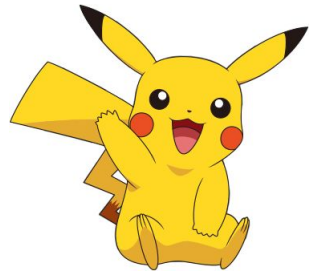
Application of Queue

2



1

Application of Queue



1

Applications

- Dijkstra's Shortest Path Algorithm using priority queue: When the graph is stored in the form of adjacency list or matrix, priority queue can be used to extract minimum efficiently when implementing Dijkstra's algorithm.
- Prim's algorithm: It is used to implement Prim's Algorithm to store keys of nodes and extract minimum key node at every step.
- Data compression : It is used in Huffman codes which is used to compresses data.

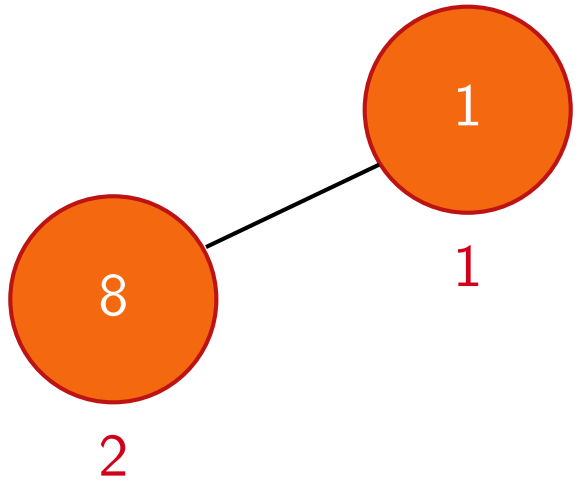
Implementations of Priority Queue

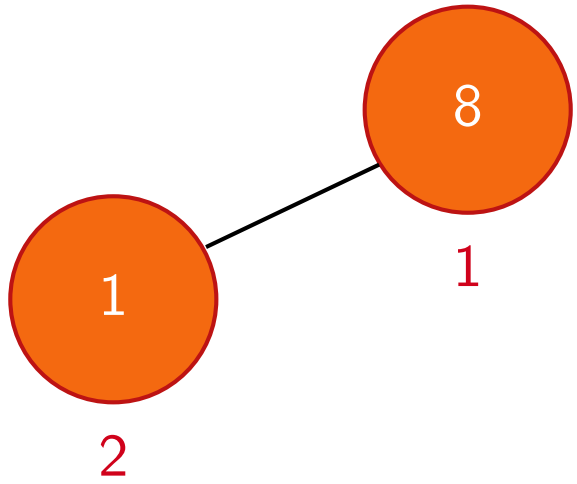
- Fibonacci Heap
- Binary Heap

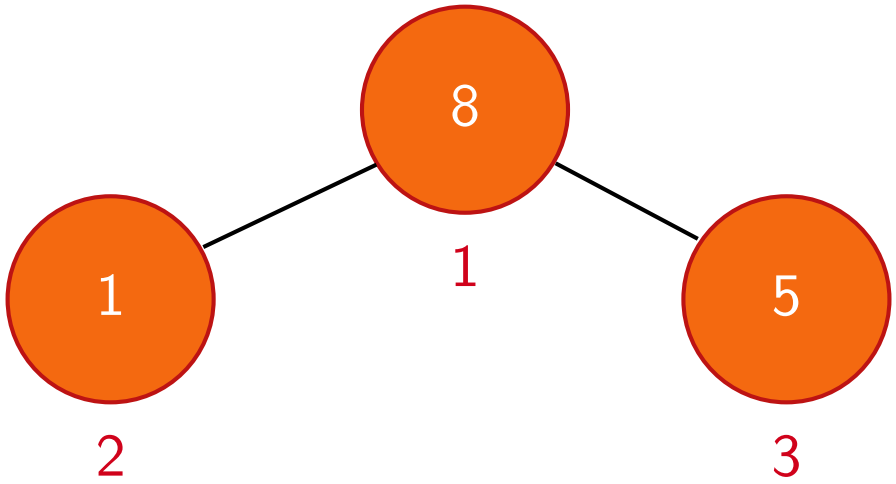
Definition

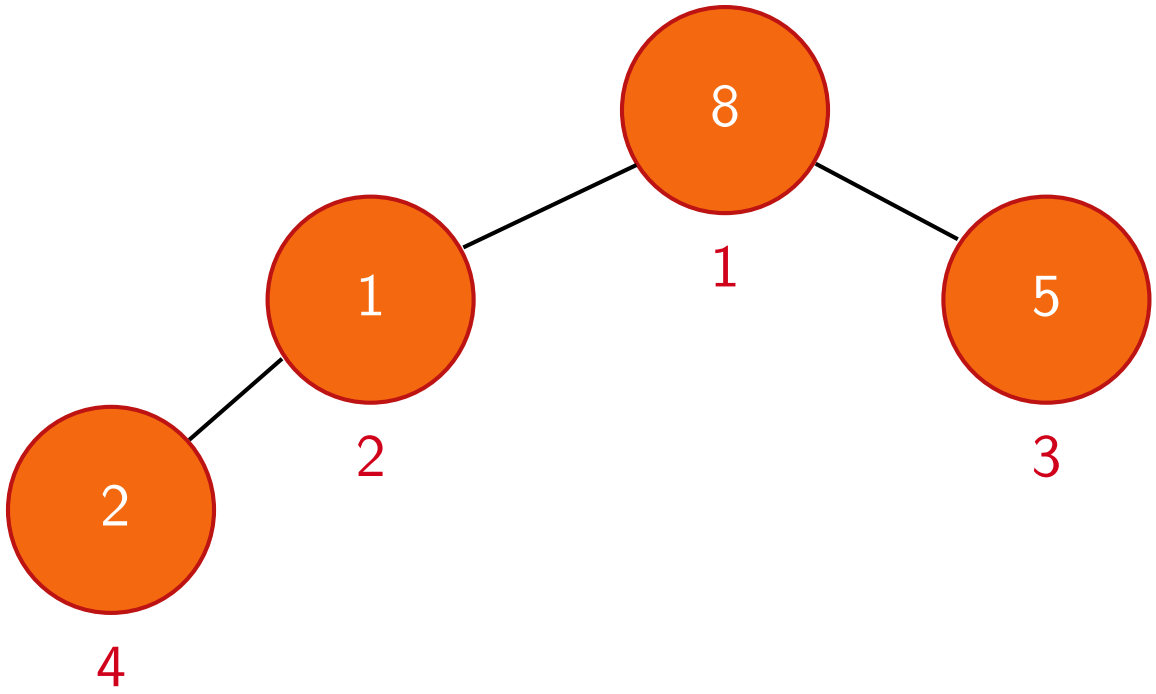
A Binary (Max) Heap is a complete binary tree that maintains the Max Heap property. Binary Heap is one possible data structure to model an efficient Priority Queue (PQ) Abstract Data Type (ADT). In a PQ, each element has a "priority" and an element with higher priority is served before an element with lower priority (ties are broken with standard First-In First-Out (FIFO) rule as with normal Queue). Try clicking ExtractMax() for a sample animation on extracting the max value of random Binary Heap above. To focus the discussion scope, we design this visualization to show a Binary Max Heap that contains distinct integers only.

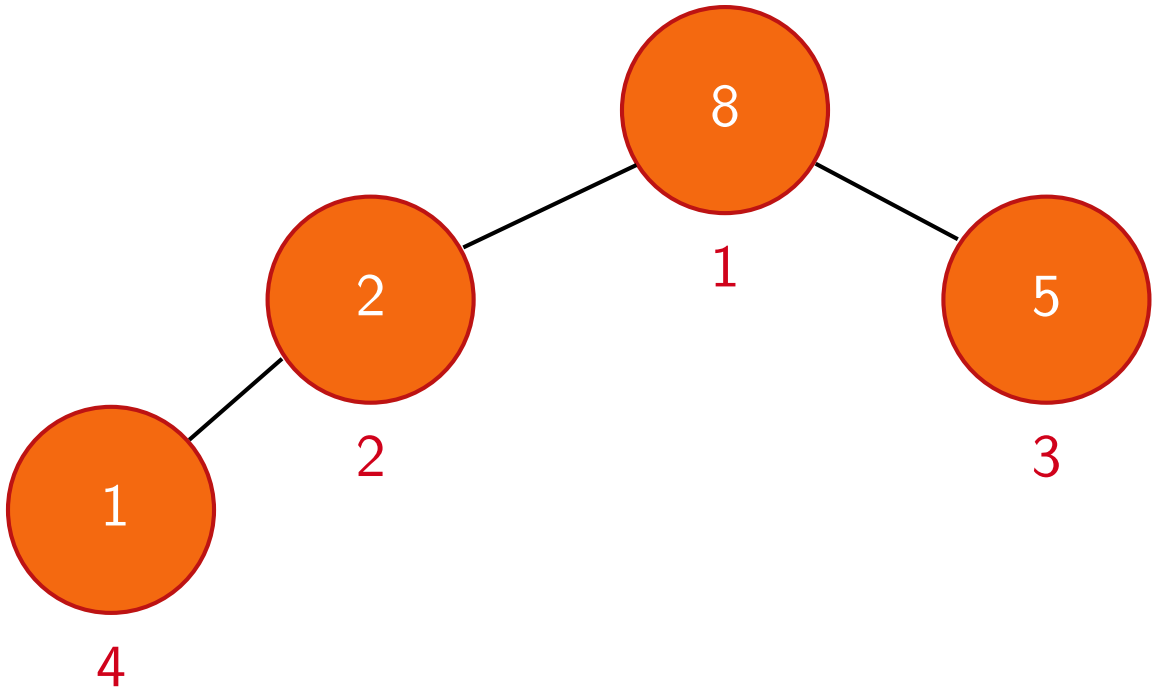
1

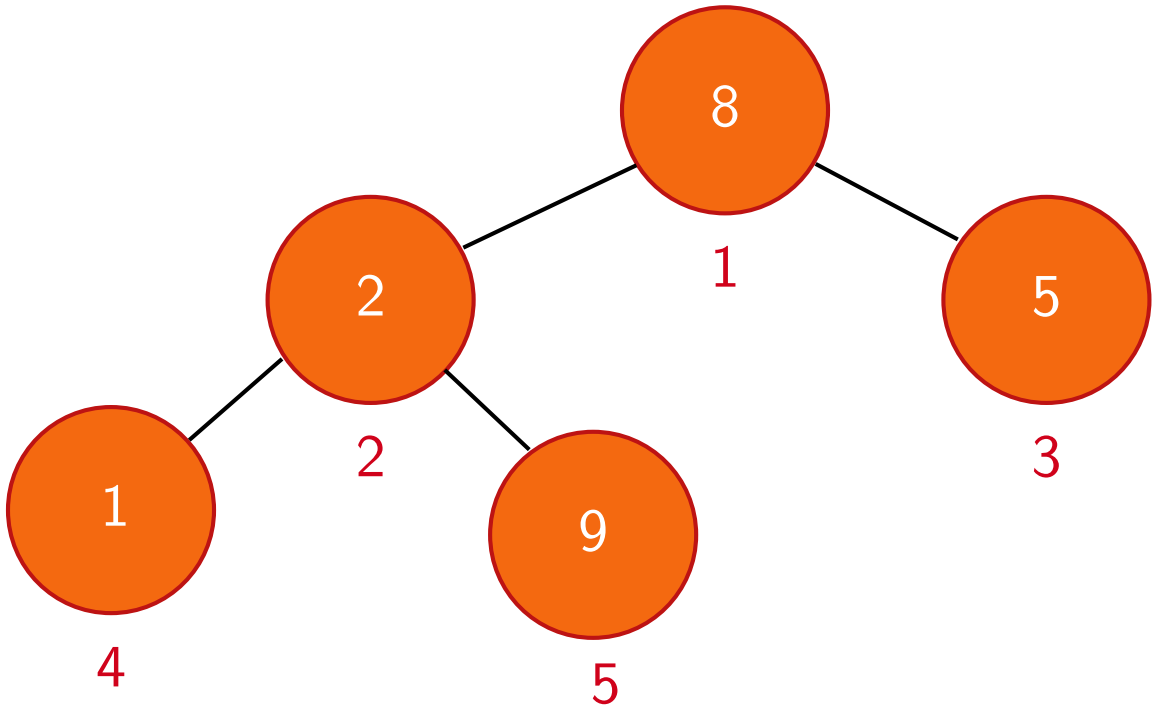

1
8
1
2

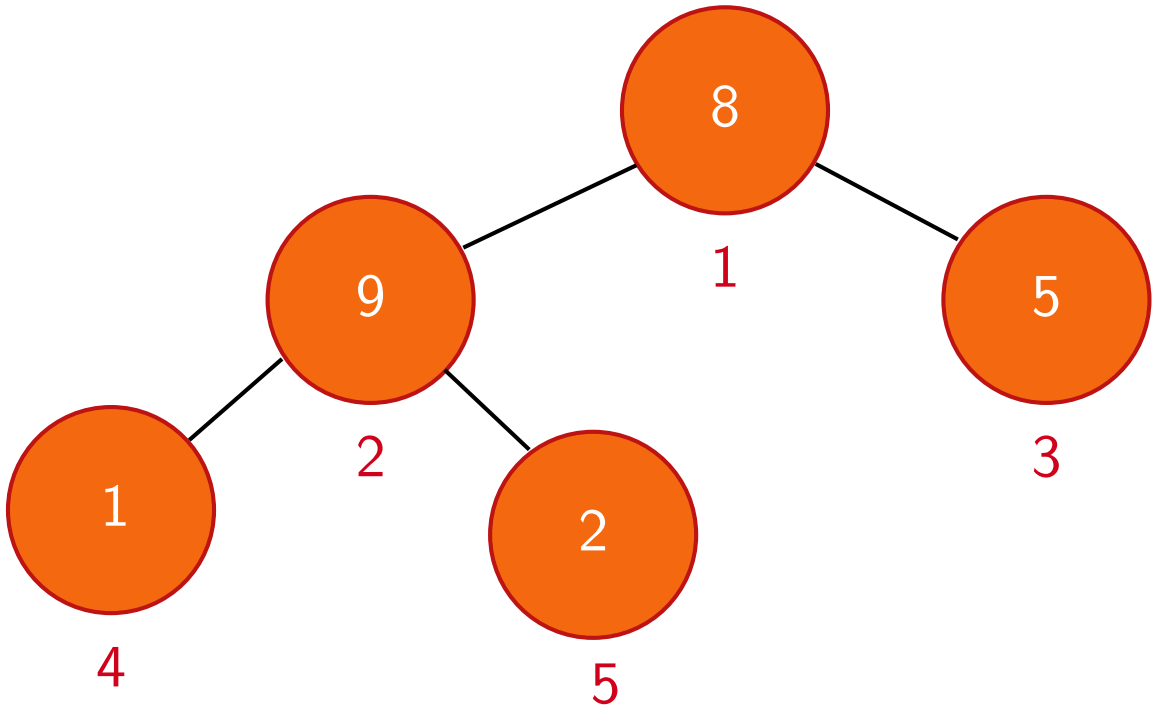

8
1
1
2

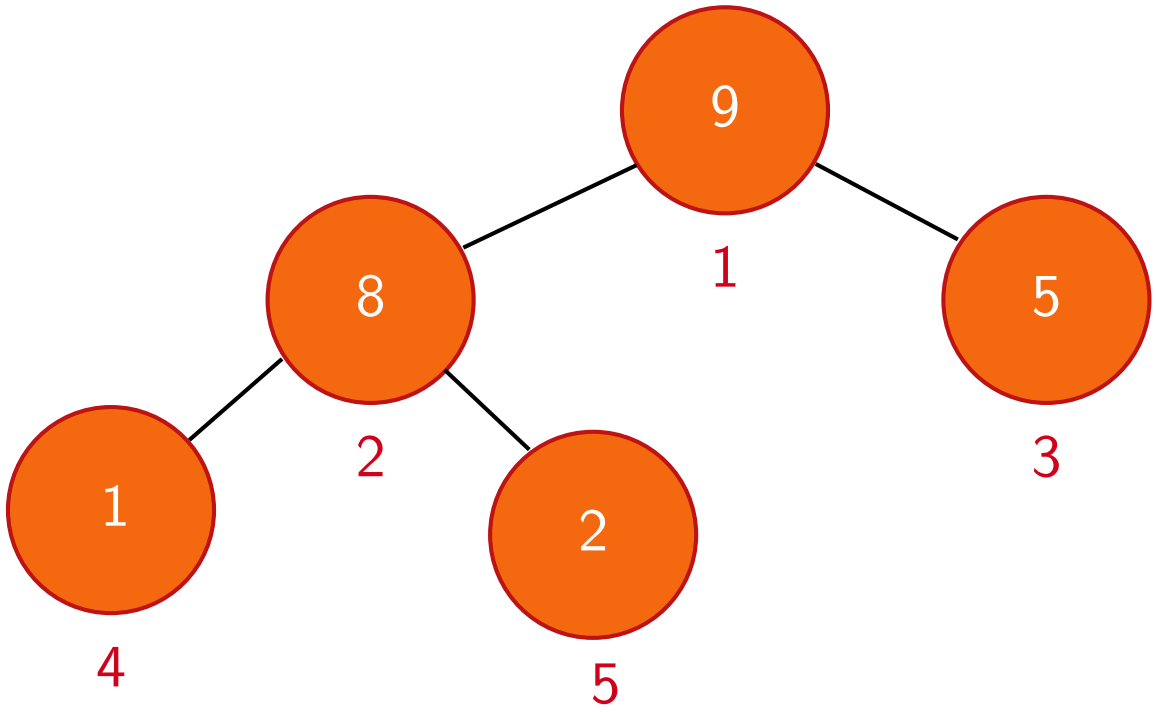

8
1
5
1
2
3

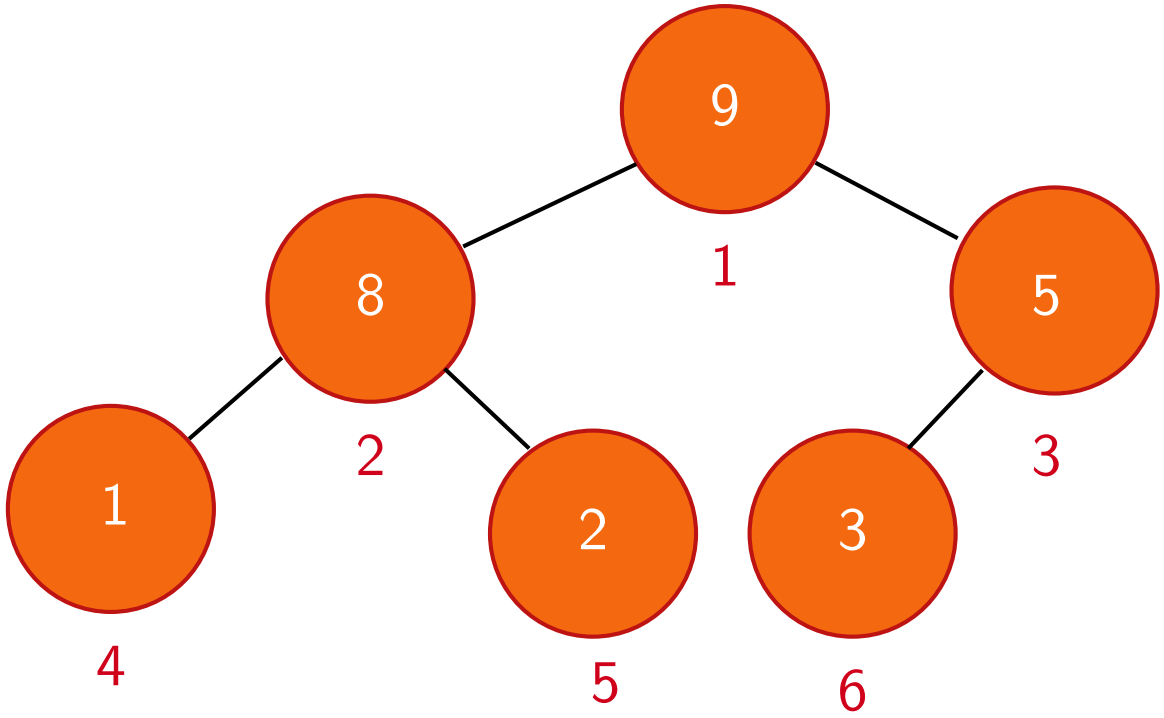

8
1
5
2
3
2
4

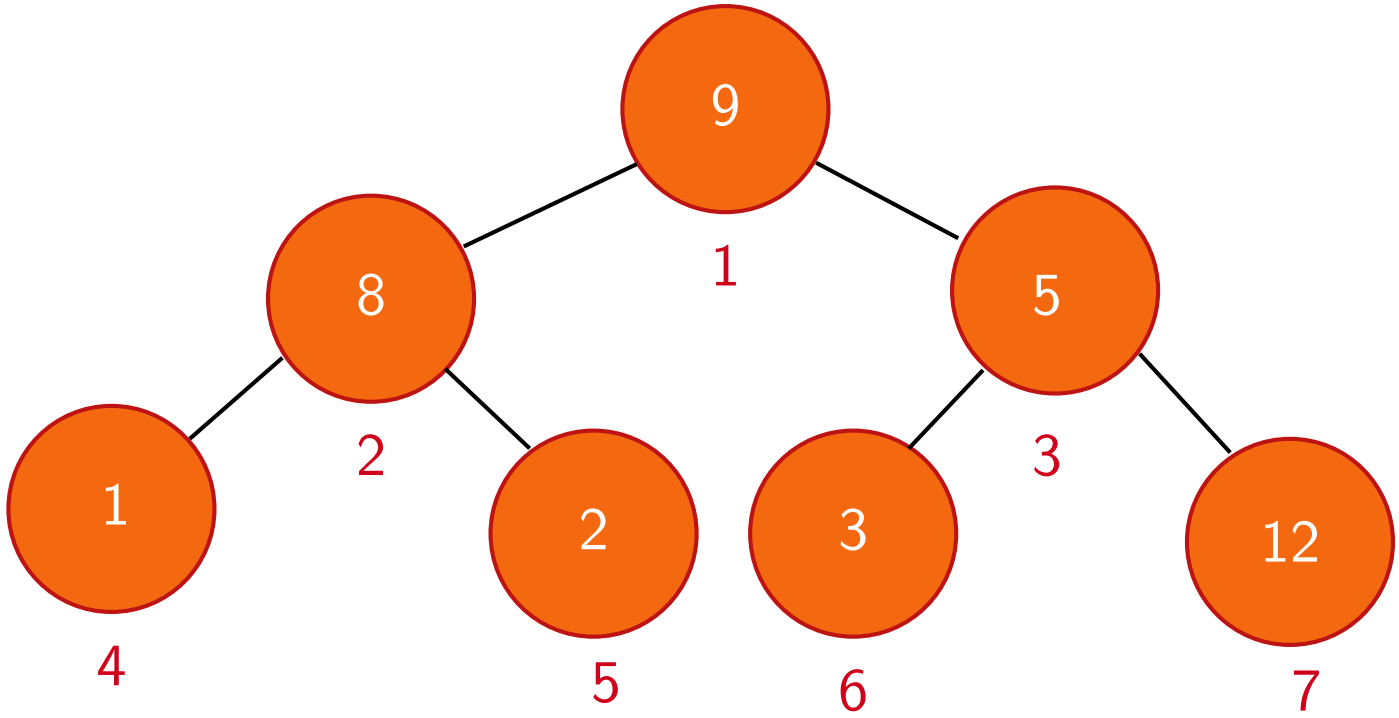

8
1
2
2
5
3
1
4

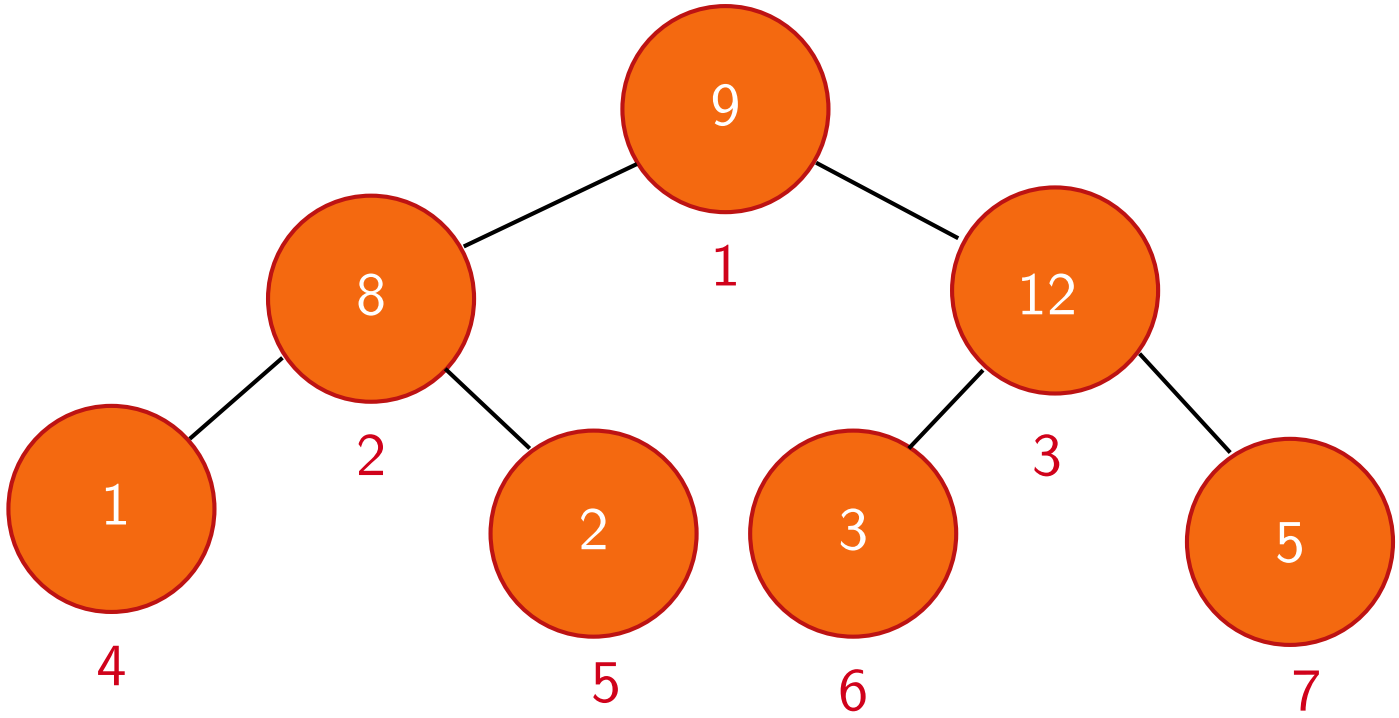

8
1
2
2
5
3
1
4
9
5

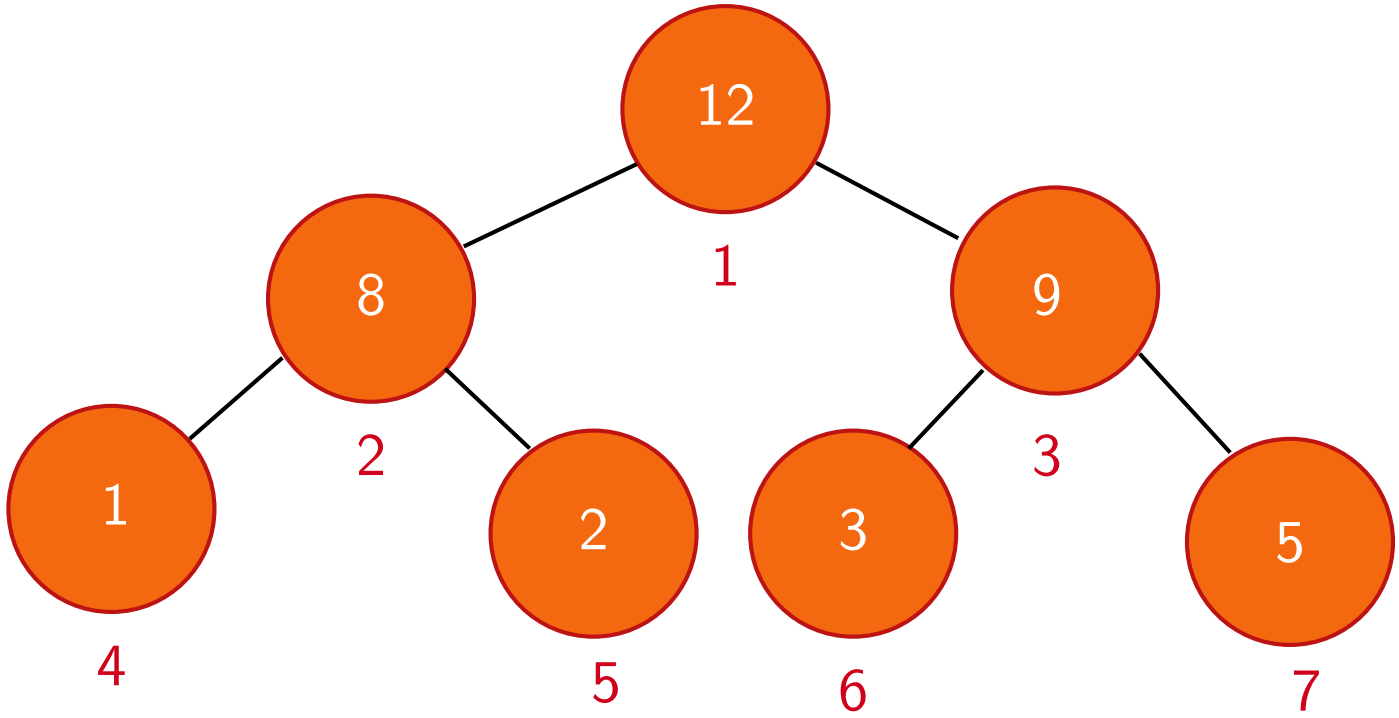

8
1
9
5
2
3
1
2
4
5


9
1
8
2
5
3
1
4
2
5


9
1
8
2
5
3
1
4
2
5
3
6


9
1
8
5
2
3
1
2
3
12
4
5
6
7


9
1
8
12
2
3
1
2
3
5
4
5
6
7


12
1
8
9
2
3
1
2
3
5
4
5
6
7

Max-Heapify Algorithm

MAX-HEAPIFY(A,i)

$l \leftarrow$ LEFT(i)
$r \leftarrow$ RIGHT(i)
if $l \leq heap\_size[A]$ *and* $A[l] > A[i]$ **then**
  $largest \leftarrow l$
else
  $largest \leftarrow i$
end if
if $r \leq heap\_size[A]$ *and* $A[r] > A[largest]$ **then**
  $largest \leftarrow r$
end if
if $largest \neq i$ **then**
  exchange $A[i] \leftrightarrow A[largest]$
end if
MAX-HEAPIFY(A,$largest$)

The End

Any Questions?