Lecture 12 - Priority Queue and Heap
Scheduling

- In a multi-user computer systems, multiple users submit jobs to run on a single processor.
- We assume that the time required by each job is known in advance. Further jobs can be preempted (stopped and resumed later).
- One policy which minimizes the average waiting time is SRPT (shortest remaining processing time).
Scheduling

- The processor schedules the job with the smallest remaining processing time.
- If while a job is running, a new job arrives with processing time less than the remaining time of current job, then the current job is preempted.
Data Structure for SRPT

- We need to maintain the remaining processing time of the unfinished jobs at any point of time.
- We need to find the job with the shortest remaining processing time.
- When a job finishes, we should remove it from our collection.
- When a new job arrives, we need to add it to the collection.
Priority Queues

A priority queue is an ADT for maintaining a set \( S \) of elements, each with an associated value called priority.

A PQ supports the following operations:

- **Insert\((x)\):** insert element \( x \) in set \( S \) \((S \leftarrow S \cup x)\)
- **Minimum():** returns the element of \( S \) with smallest priority.
- **Delete-min():** returns and removes the element of \( S \) with smallest priority.
Priority Queue

Applications

- job scheduling shared computing resources (Unix)
- Event simulation
- As a building block for other algorithms

A Heap can be used to implement a PQ.
Why do we need a Heap?

Unsorted array:
- Insert takes $O(1)$
- Search takes $O(n)$
- Find min takes $O(n)$
- Delete min takes $O(n)$
Why do we need a Heap?

Sorted array:
- Insert takes $O(n)$
- Search takes $O(\log(n))$
- Find min takes $O(1)$
- Delete min takes $O(n)$
Why do we need a Heap?

- Unsorted linked list:
  - Insert takes $O(1)$
  - Search takes $O(n)$
  - Find min takes $O(n)$
  - Delete min takes $O(n)$
Why do we need a Heap?

- Heap:
  - Insert takes $O(\log(n))$
  - Find min takes $O(1)$
  - Delete min takes $O(\log(n))$
What is a binary heap?

- Each node has atmost two children.
- Complete binary tree or atmost complete binary tree qualified as binary heap
- Node with no children is also qualified as heap.
- Left skewed or right skewed tree not a heap.
- Max heap and Min heap
- Let us look at some examples:
What are the other heaps?

- 3 ary heap
- N ary heap
Implementation of heap?

- Can we use array to implement a Heap?
Questions