

BCSE202L Data Structures and Algorithms

Priority Queues

Applications of Heaps

- One excellent application of heap is heapsort which often beats quicksort and mergesort
- Another utility is the implementation of priority queue
- There are max priority queues and min priority queues
- We will focus on max priority queues which are implemented using max heaps

Priority Queues

- A variation of queue
- Each element has a priority value associated with it
- Priority queue is a type of queue that arranges elements based on their priority values
- An element with high priority is dequeued before an element with low priority.
- If two elements have the same priority, they are dequeued according to their order in the queue.

Types of priority queues

- Max Priority Queue (Descending Order Priority Queue): In max priority queue, the element with a higher priority value is given a higher priority in the priority list (Eg. Mark)
- Min Priority Queue (Ascending Order Priority Queue): In min priority queue, the element with a lower priority value is given a higher priority in the priority list (Eg. Rank)

Applications of priority queue

- Job scheduling on a computer (Max PQ)
- Event driven simulator (Min PQ)
- OS related tasks: priority scheduling, load balancing and interrupt handling
- Implementation of other algorithms: Dijkstra's, Prim's
- Data compression techniques like Huffman coding

Array/Linked List Implementation

- In a normal queue, both enqueue and dequeue takes $O(1)$ time
- In priority queue, if enqueue is done in the arriving order of elements, it takes $O(1)$ time
- But then dequeue will take $O(n)$ time as the priorities are to be compared and the one with the highest priority is to be dequeued

Array/Linked List Implementation

- Alternatively, if enqueue is done in such a way that higher priority elements should go to the front, then it takes $O(n)$ time to enqueue
- In this case, dequeue will take $O(1)$ only as higher priority elements are towards the front

Priority Queue Implementation

- Array/Linked list implementation of priority queue is not ideal because either enqueue or dequeue would take $O(n)$ time
- The best data structure to implement priority queue is

Priority Queue Implementation

- Array/Linked list implementation of priority queue is not ideal because either enqueue or dequeue would take $O(n)$ time
- The best data structure to implement priority queue is **heap**
- Why?

Priority Queue Implementation

- Array/Linked list implementation of priority queue is not ideal because either enqueue or dequeue would take $O(n)$ time
- The best data structure to implement priority queue is **heap**
- Why? Insertion takes $O(\log n)$ time and deletion takes $O(1)$ time
- Max heap can be used to implement max priority queue and min heap for min priority queue

Max PQ using Max Heaps

- Construct a heap with priority values
- Perform deletion one-by-one
- Recall MAX-HEAPIFY,
BUILD-MAX-HEAP,
EXTRACT-MAX/DELETE from the Heaps
classes
- Similarly implement Min PQ using Min Heap

Thank you..!!