DATA STRUCTURES AND ALGORITHMS

Priority Queue

By
Zainab Malik

Content

- Introduction to Priority Queue
- Representation of Priority Queue
  - Representation using Linked list
  - Representation using Multiple Queue
  - Representation using heap (will be covered in detail later)

Priority Queue

- A priority queue is a kind of queue in which each element is assigned a priority
- The order in which elements are processed comes from the following rules:
  - An element with highest priority is processed first before any element of lower priority.
  - Two or more elements with the same priority are processed according to the order in which they are added to the queue.

Priority Queue

- There are different scenarios in which we process information/data on the basis of some assigned priorities e.g.
  - In process scheduling, shortest job may given a higher priority over the long one which means that shortest job will be executed first.
  - In NADRA or passport office we apply for urgent processing of our application by paying additional charges so that our application get higher priority as compared to normal applications.
    - For all urgent applications FIFO or LILO order will still be maintained.

5

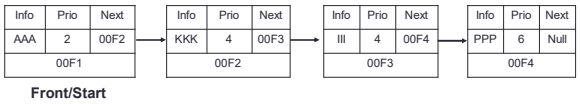
Representation of Priority Queue

- There are various ways of representing (maintaining) a priority queue in memory
  - Using a Linked List
  - Using multiple queues, one for each priority
  - Using a heap (another data structure)

6

Priority Queue using Linked List

- In this representation, each node of the linked list has three slots/fields
  - An information field that holds the element of the queue
  - A priority field that holds the priority number of the element
  - A next filed that holds the address of next node in the priority queue



Front/Start

Note: In this example lower number is considered as higher priority value

7

Priority Queue using Linked List

- **Enqueue(item, priority):** The elements will be enqueued in a sorted manner
  - Before adding an element we need to find its correct position by comparing the priority number, then the node well be added at identified position
  - Same as addSorted() function of a singular Linked List
- **Dequeue():** As the node with highest priority is kept at the beginning therefore, to process on the basis of priority we need to remove node from the beginning.
  - Same as removeFromBegining() function of a singular Linked List

8

Priority Queue using Linked List

- Advantages:
  - Elements to be processed first is kept at the beginning therefore, it is very easy to delete and process element with the highest priority.
- Disadvantages:
  - As the linked priority queue maintains the sorted list of elements (on the basis of priority) therefore, insertion of the elements will be time consuming because before insertion we need to find its correct position first.

9

Priority Queue using Multiple Queues

- In this representation, one separate queue is maintained for each priority number.

Priority Queue

	front	rear
0	0	1
1	-1	-1
2	1	3
3	0	2

	0	1	2	3	4	5
0	aa	bb				
1						
2		pp	qq	rr		
3	tt	uu	vv			

- This priority queue has 4 different levels of priorities (0,1,2,3) and maintained by mean of four separate queues
- Each queue has its own rear and front

10

Priority Queue using Multiple Queues

- Enqueue(item, priority): the element is added at rear end of provided priority queue
  - e.g. enqueue("ss", 2): "ss" will be added at index 4 of queue (2)
- Dequeue(): Element from the first non-empty highest priority queues is processed/removed first
  - e.g. dequeuer(): aa will be removed first and then bb, pp,qq,rr,tt,uu,vv in subsequent dequeuer calls

Priority Queue

	front	rear
0	0	1
1	-1	-1
2	1	3
3	0	2

	0	1	2	3	4	5
0	aa	bb				
1						
2		pp	qq	rr		
3	tt	uu	vv			

11

Priority Queue using Multiple Queues

Priority Queue

	front	rear
0	0	1
1	-1	-1
2	1	3
3	0	2

	0	1	2	3	4	5
0	aa	bb				
1						
2		pp	qq	rr		
3	tt	uu	vv			

- Dequeue(): Which element would be processed/removed?
- Enqueue(cc,0): Where it will be inserted?
- Enqueue(ww,2): Where it will be inserted?
- Dequeue(): Which element would be processed/removed?
- Dequeue(): Which element would be processed/removed?
- Dequeue(): Which element would be processed/removed?

12

Priority Queue using heap

- Will be discussed after discussion on heap data structure

13

Assignment

- Implement Priority queue using Multiple queues
  - Ask level of priorities from User
  - Declare an array of size according to level of priorities
  - Each slot of this array must contain an object of Queue Class

14

Thank You