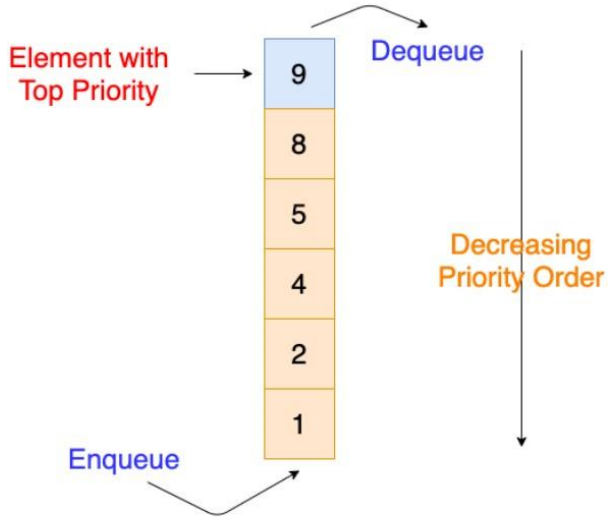
Implement Priority Queue Using Linked list

THEORY:

A priority queue is a special type of queue in which each element is associated with a priority and is served according to its priority. If elements with the same priority occur, they are served according to their order in the queue.

Generally, the value of the element itself is considered for assigning the priority. For example, The element with the highest value is considered as the highest priority element. However, in other cases, we can assume the element with the lowest value as the highest priority element. In other cases, we can set priorities according to our needs.



Difference between Priority Queue and Normal Queue is that, In a queue, the first-infirst-out rule is implemented whereas, in a priority queue, the values are removed on the basis of priority. The element with the highest priority is removed first.

Algorithm :- Insert operation :-

1. IF((Front == 0)&&(Rear == N-1))
2. PRINT “Overflow Condition”
3. Else
4. IF(Front == -1)
5. Front = Rear =0
6. Queue[Rear] = Data
7. Priority[Rear] = Priority
8. ELSE IF(Rear ==N-1)
9. FOR i=Front;i<=Rear;i++)
10. FOR(i=Front;i<=Rear;i++)
11. Q[i-Front] =Q[i]
12. Pr[i-Front] = Pr[i]
13. Rear = Rear-Front
14. Front = 0
15. FOR(i = r;i>f;i–)
16. IF(p>Pr[i])
17. Q[i+1] = Q[i] Pr[i+1] = Pr[i]

18. ELSE
19. Q[i+1] = data Pr[i+1] = p
20. Rear++

Delete operation :- 1.
IF(Front == -1)
2. PRINT “Queue Under flow condition”
3. ELSE
4. PRINT”Q[f],Pr[f]”
5. IF(Front==Rear)
6. Front = Rear = -1
7. ELSE
8. FRONT++

SOURCE CODE:

```
#include <iostream>
using namespace std;
struct Node
{
   int data;
   int priority;
   struct Node *next;
};
Node *front = NULL;
Node *rear = NULL;

int Size = 0;
void insert(int val, int pr)
{
   Node *q;
   Node *temp = new Node;
   temp->data = val;
   temp->priority = pr;
   if (front == NULL || pr < front->priority)
   {
      temp->next = front;
      front = temp;
   }
   else
   {
      q = front;
      while (q->next != NULL && q->next->priority <= pr)
      {
         q = q->next;
      }
      temp->next = q->next;
      q->next = temp;
   }
```

```
   Size++;
}

void del()
{
   Node *temp;
   if (front == NULL)
   {
      cout << "Priority Queue is empty" << endl;
   }
   else
   {
      temp = front;
      cout << "Element deleted from priority queue is: " << temp->data << endl;
      front = front->next;
      delete temp;
      Size--;
   }
}

void display()
{
   Node *ptr;
   ptr = front;
   if (front == NULL)
   {
      cout << "Priority Queue is empty" << endl;
   }
   else
   {
      cout << "Priority Queue is: \n";
      cout << "Priority      Item\n";
      while (ptr != NULL)
      {
         cout << ptr->priority << "\t" << ptr->data << "\n";
         ptr = ptr->next;
      }
   }
}

int main()
{
 int choice, n, pr;
   int val;
   cout<<"PRIORITY QUEUE OPERATIONS"<<endl;
   cout<<"\n-----------------------------------------------"<<endl;
   cout<<"\n 1.INSERT\n 2.DELETE\n 3.DISPLAY\n 4.EXIT"<<endl;
   do
   {
      cout<<"Enter the Choice: \n"<<endl;
      cin>>choice;
```

```
    switch (choice)
    {
    case 1:
       cout << "Enter the value: ";
       cin >> n;
       cout << "Enter the priority: ";
       cin >> pr;
       insert(n, pr);
       break;
    case 2:
       del();
       break;
    case 3:
       display();
       break;
    case 4:
       exit(0);
    default:
        cout<<"Invalid Input!, Please Enter a Valid Choice(1/2/3/4) \n"<<endl;
    }
  }
  while(choice!=4);

  return 0;
}
```

OUTPUT:

```
Select "C:\Users\NARENDER KESWANI\Documents\FYMCA\DSA\priorityQueue.exe"
PRIORITY QUEUE OPERATIONS

-----------------------------------------------

 1.INSERT
 2.DELETE
 3.DISPLAY
 4.EXIT
Enter the Choice:

3
Priority Queue is empty
Enter the Choice:

2
Priority Queue is empty
Enter the Choice:

1
Enter the value: 5
Enter the priority: 6
Enter the Choice:

1
Enter the value: 7
Enter the priority: 1
Enter the Choice:

3
Priority Queue is:
Priority        Item
1       7
6       5
Enter the Choice:

3
Priority Queue is:
Priority        Item
1       7
6       5
Enter the Choice:

2
Element deleted from priority queue is: 7

Enter the Choice:

3
Priority Queue is:
Priority        Item
6       5
```

CONCLUSION:

From this practical, I have learned from priority queue implementation.