C, C++, DSA in depth

priority_queue

Saurabh Shukla (MySirG)

Agenda

1. priority_queue
2. Creating priority_queue object
3. priority_queue methods
4. priority_queue for non primitive types.

priority_queue

A template container adaptor class that provides a restriction of functionality limiting access to the top element of some underlying container type, which is always the largest or the highest priority.

priority_queue

New elements can be added to the priority_queue and the top element of the priority_queue can be inspected or removed.

Suitable underlying container classes for priority_queue include deque class and the default vector class or any other sequence container that supports the operations of front, push_back and pop_back and a random access iterator.

priority_queue

The underlying container class is encapsulated within the container adaptor, which exposes only the limited set of the sequence container member functions as a public interface.

The header for the STL priority_queue library is <queue>

How to create a priority_queue object?

```
priority_queue <int, deque<int>> q1;
priority_queue <int, vector<int>> q2;
```

Methods of priority_queue

empty()

pop()

push()

size()

top()

How to create a priority_queue for non-primitive type?

```
class Employee {
        private:
                int empid;
                string name;
                float salary;
        public:
                Employee(int i, string n, float s) : empid(i), name(n), salary(s)
                { }
                float getSalary ()
                { return salary; }
};
```

Create priority_queue on the basis of Salary.

```
class compareSalary {
    public:
      bool operator()(Employee &e1, Employee &e2)
      {
          return e1.getSalary() > e2.getSalary();
      }
};

priority_queue<Employee, vector<Employee>, compareSalary> pq;
```