Data Structures and Algorithms

Prof. Ajit A. Diwan
Prof. Ganesh Ramakrishnan
Prof. Deepak B. Phatak
Department of Computer Science and Engineering
IIT Bombay

Session: Queues using STL

The Queue Class

- First in First Out (Push at back, Pop from front)
- Container Adaptor
- Created using existing container.
- Must use **#include<queue>** at the beginning of the program

The Queue Class: Member Functions

- Member Functions
  - empty()
  - size()
  - front()
  - back()
  - push()
  - pop()
  - swap()

Program 1: Basic concepts of queue

```
#include <iostream>
#include <queue>
using namespace std;
int main ()  {
    queue<char> q1;
     // In this example, we will see how to
    // push three characters ‘a’, ‘b’ and ‘c’
    // on a queue
    return 0;
}
```

Program 1: Queue (Contd.)

```
int main ()  {
  queue<char> q1;
  int i=0;     char element='a';
  if(q1.empty())
    cout << "Queue is empty as expected"  << endl;
  else
    cout << "Queue is not empty" << endl;
  for(i=0;i<=2;i++) {
    q1.push(element);
    cout<<"Element "<<element<<" pushed in"<<endl;
    element++; // successive  ‘next’ character
  }
  cout << "Size of queue: "  << q1.size() << endl;
```

Empty queue of type char

Since Queue is empty, “Queue is empty as expected” is printed

Element a pushed in
Element b pushed in
Element c pushed in

Size of queue: 3

Program 1: Queue (Contd.)

```
Front element: a,
Back element: c
Element popped
Front element: b,
Back element: c
Element popped
Front element: c,
Back element: c
Element popped
```

Since Queue is empty, “Queue is empty” is printed

```
while(!q1.empty()) {
   cout << "Front element: " << q1.front()
         << ", Back element: " << q1.back() << endl;
   q1.pop();
   cout << "Element popped" << endl;
}

if(q1.empty())
   cout << "Queue is empty" << endl;
else
   cout << "Queue is not empty" << endl;
return 0;
}
```

Points to remember

Never do the following:

- Accessing ‘front’ or ‘back’ when queue is empty
- Popping elements when queue is empty

Queue: swap()

- Swaps contents of two queues
- Should be of the same type
- Sizes may be different

```
int main ()  {
    queue<int> q1, q2;
    int i=0;
    for(i=0;i<=2;i++)  {   q1.push(i);  }
    for(i=0;i<=5;i++)  {   q2.push(i);  }
    cout << "Size of queues : "  << q1.size() << "\t"  << q2.size() << endl;
    swap(q1,q2);
    cout << "Size of queues : "  << q1.size() << "\t"  << q2.size() << endl;
    return 0;
}
```

Take-home activity

Practice problem

- Assume that the function swap() is not available in STL
  - Write your own swap function, and test the above program using it

References

- https://en.wikipedia.org/wiki/Queue_(abstract_data_type)
- http://www.cplusplus.com/reference/queue/queue/

Thank you