Lecture 31 Queues

1. Stl Implementation
2. #include <iostream>
3. #include <queue>
4. using namespace std;
5. int main() {
6. queue<int> q;
7. for(int i=1; i<=5; i++)
8. {
9. q.push(i);
10. }
11. while(!q.empty())
12. {
13. cout<<q.front()<<"<-";
14. q.pop();
15. }
16. return 0;
17. }

2. <https://leetcode.com/problems/implement-stack-using-queues/>

class MyStack {

public:

/\*\* Initialize your data structure here. \*/

queue<int> q1;

queue<int> q2;

MyStack() {

}

/\*\* Push element x onto stack. \*/

void push(int x) {

q1.push(x);

}

/\*\* Removes the element on top of the stack and returns that element. \*/

int pop() {

if(q1.empty())

{

return -1;

}

else

{

while(q1.size()!=1)

{

int temp = q1.front();

q1.pop();

q2.push(temp);

}

int temp = q1.front();

q1.pop();

swap(q1, q2);

return temp;

}

}

/\*\* Get the top element. \*/

int top() {

if(q1.empty())

{

return -1;

}

else

{

while(q1.size()!=1)

{

int temp = q1.front();

q1.pop();

q2.push(temp);

}

int temp = q1.front();

q1.pop();

q2.push(temp);

swap(q1, q2);

return temp;

}

}

/\*\* Returns whether the stack is empty. \*/

bool empty() {

return q1.size()==0;

}

};

3. <https://leetcode.com/problems/implement-queue-using-stacks/>

class MyQueue {

public:

/\*\* Initialize your data structure here. \*/

stack<int> s1;

stack<int> s2;

MyQueue() {

}

/\*\* Push element x to the back of queue. \*/

void push(int x) {

s1.push(x);

}

/\*\* Removes the element from in front of queue and returns that element. \*/

int pop() {

if(s2.empty())

{

while(!s1.empty())

{

int temp = s1.top();

s1.pop();

s2.push(temp);

}

}

int temp = s2.top();

s2.pop();

return temp;

}

/\*\* Get the front element. \*/

int peek() {

if(s2.empty())

{

while(!s1.empty())

{

int temp = s1.top();

s1.pop();

s2.push(temp);

}

}

int temp = s2.top();

return temp;

}

/\*\* Returns whether the queue is empty. \*/

bool empty() {

return (s1.empty() && s2.empty());

}

};

4. First Non Repeating Character

#include <iostream>

#include <queue>

using namespace std;

int main() {

    queue<char> q;

    int freq[27] = {0};

    char ch;

    cin>>ch;

    while(ch!='.')

    {

        q.push(ch);

        freq[ch-'a']++;

        while(!q.empty())

        {

            int idx = q.front() - 'a';

            if(freq[idx] > 1)

            {

                q.pop();

            }

            else

            {

                cout<<q.front()<<" ";

                break;

            }

        }

        if(q.empty())

        {

            cout<<"-1"<<" ";

        }

        cin>>ch;

    }

    return 0;

}