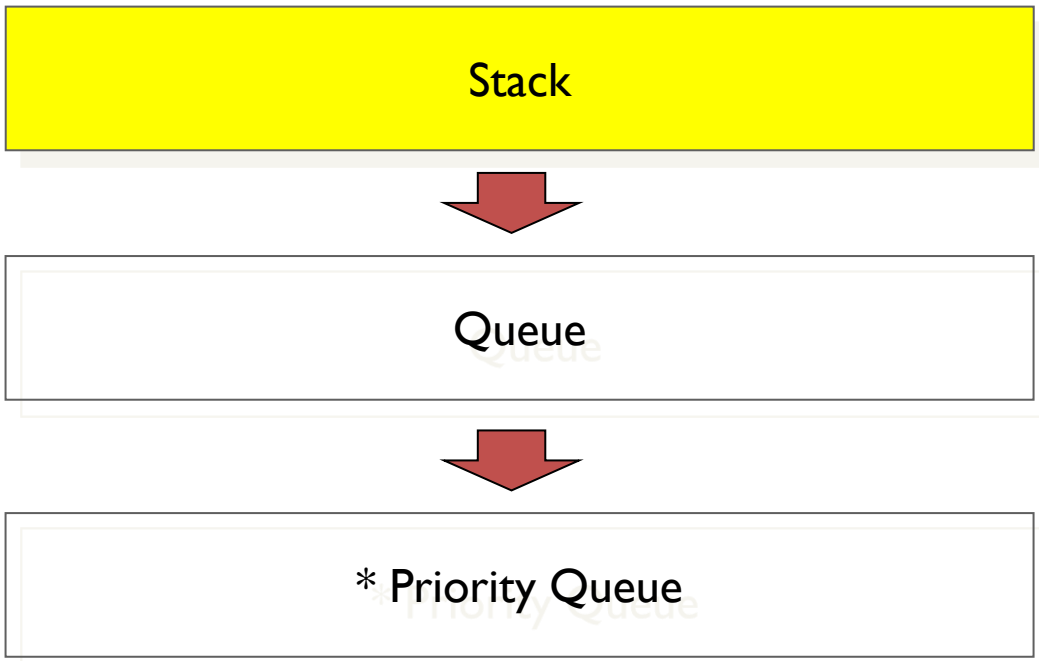
NCKU Programming Contest Training Course
Stack, Queue, Priority Queue
2018/02/24

Yu-Cheng Chang (Vic)
vic85821@gmail.com

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

made by electron & free999

Outline



made by electron & free999

Stack

- Stack
  - A stack is an ordered list in which insertions and deletions are made at one end called the top.
  - If we add the elements *A*, *B*, *C*, *D*, *E* to the stack, in that order, then *E* is the first element we delete from the stack
  - A stack is also known as a ***Last-In-First-Out (LIFO)*** list.

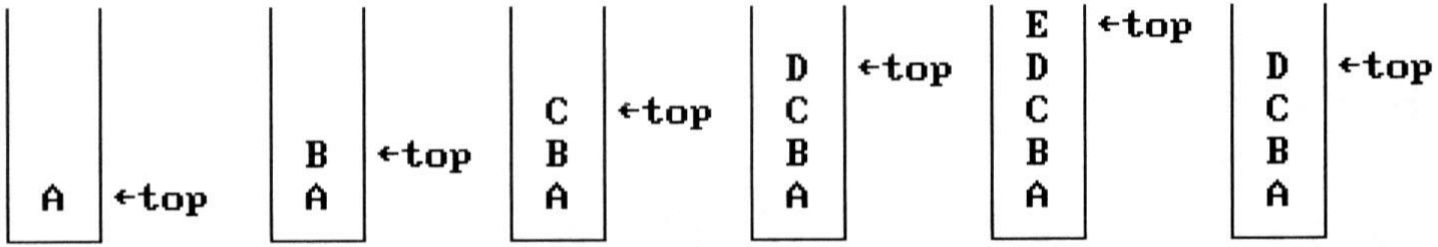
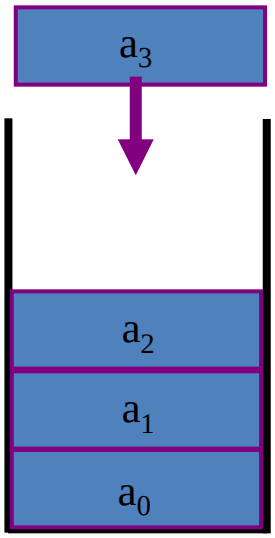


Figure 3.1: Inserting and deleting elements in a stack

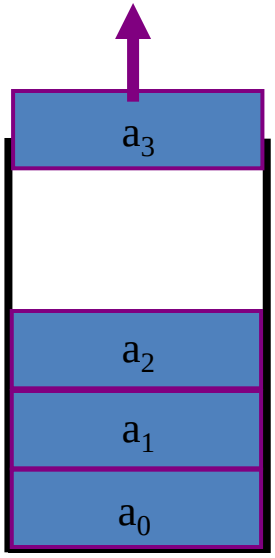
made by electron & free999

Stack

- Member Function
  - push
  - pop
  - top
  - empty
  - size



Push (Add)



Pop (Delete)

made by electron & free999

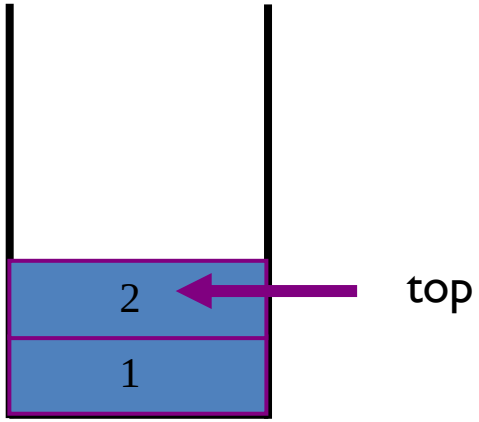
Stack

- Stack Usage in STL – Standard Template Library

```
/* stack example */
#include <iostream>
#include <stack>
using namespace std;

int main()
{
        stack<int> stk;
        stk.push(1);
        stk.push(2);
        cout<< stk.top();  // 2
        cout<< stk.empty(); // false

        /* clear the stack */
        while(!stk.empty()) stk.pop();
}
```



made by electron & free999

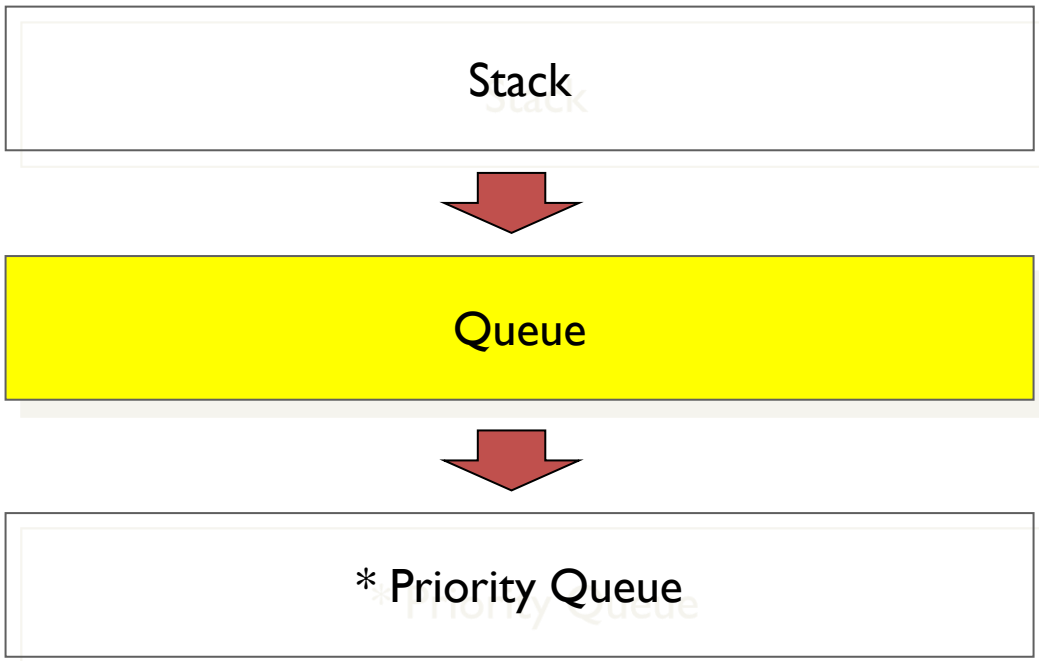
Practice 1

Uva-673

Parantheses

made by electron & free999

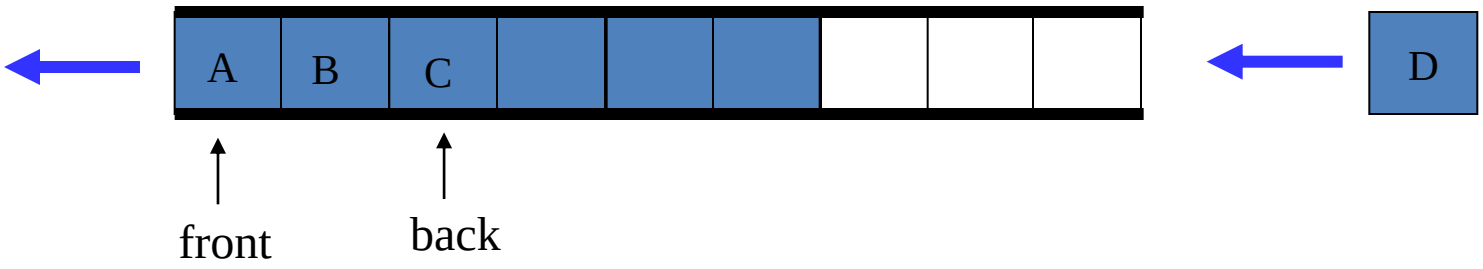
Outline



made by electron & free999

Queue

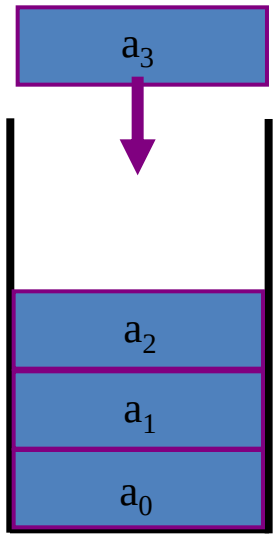
- Queue
  - A queue is an ordered list in which insertions and deletions are made at one end called the front
  - If we add the elements *A*, *B*, *C*, *D*, *E* to the stack, in that order, then *A* is the first element we delete from the queue
  - A stack is also known as a ***First-In-First-Out (FIFO)*** list.



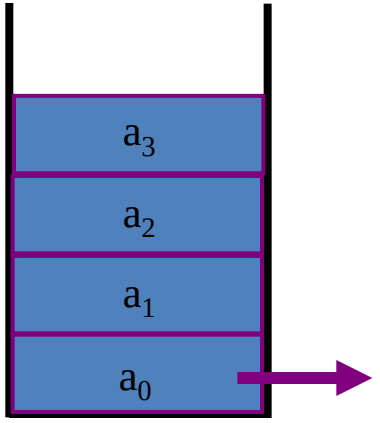
made by electron & free999

Queue

- Member Function
  - push
  - pop
  - front
  - back
  - empty
  - size



Push (enqueue)



Pop (dequeue)

made by electron & free999

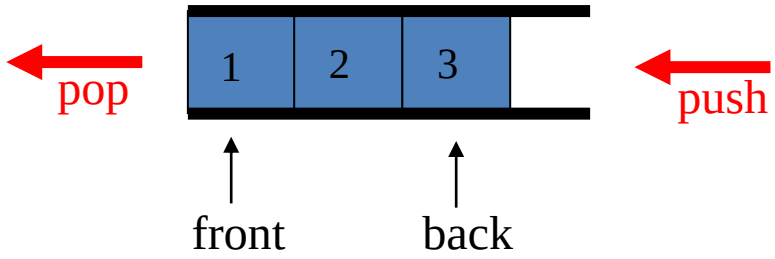
Queue

- Queue Usage in STL

```
/* stack example */
#include <iostream>
#include <queue>
using namespace std;

int main()
{
        queue<int> que;
        que.push(1);
        que.push(2);
        que.push(3);
        cout<<que.front(); // 1

        /* clear the stack */
        while(!que.empty()) que.pop();
}
```

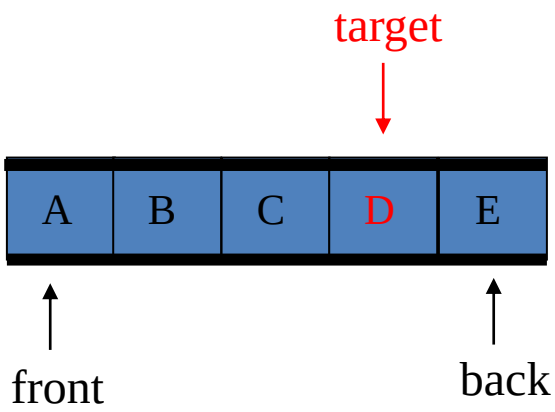


made by electron & free999

Queue

- Scan elements in queue

```
while(que.front() != 'D') {
        que.pop();
}
```



made by electron & free999

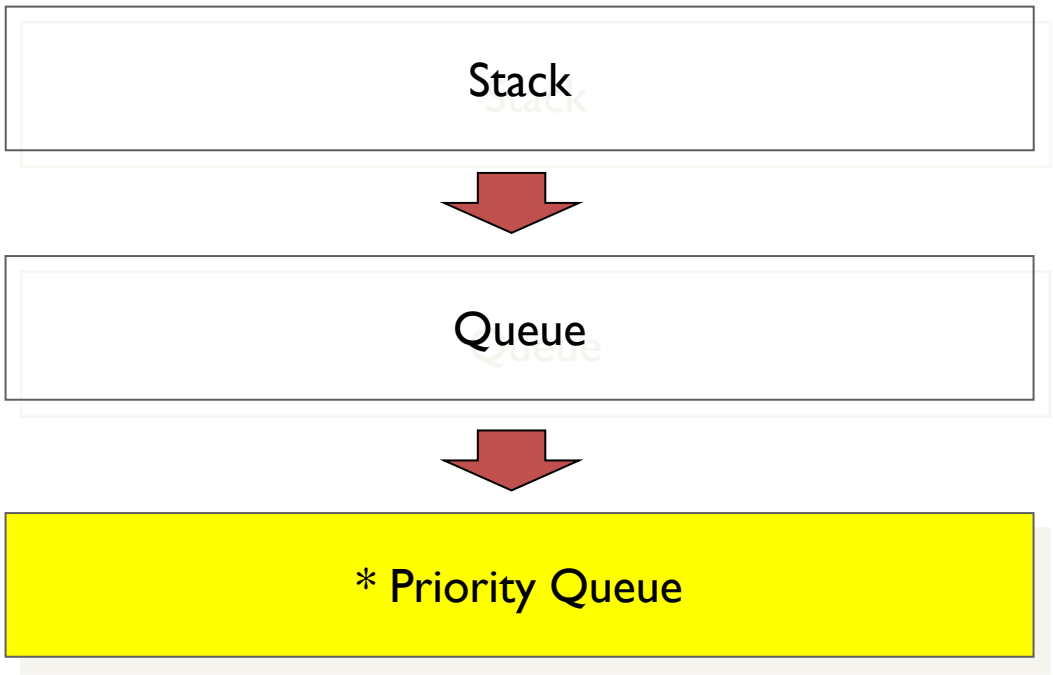
Practice 2

POJ – 3125

Printer Queue

made by electron & free999

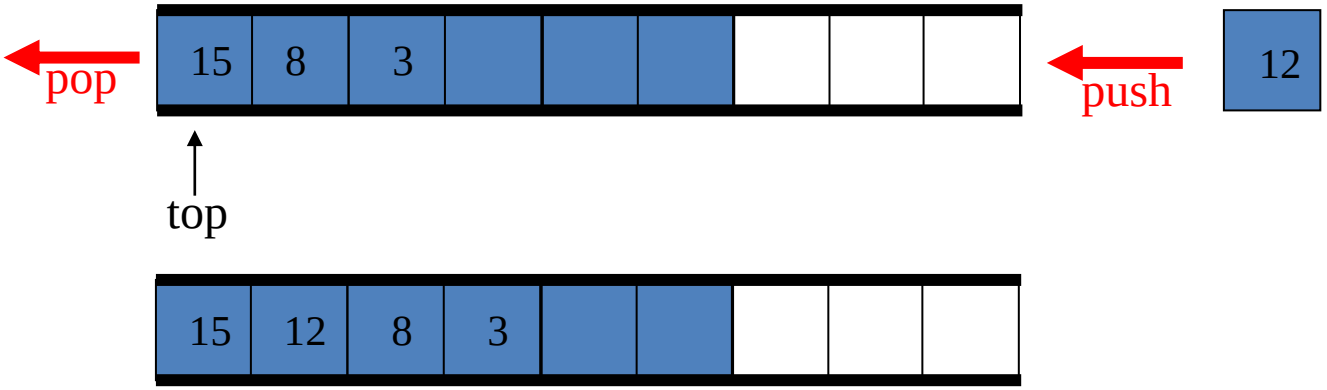
Outline



made by electron & free999

(補充) Priority Queue

- Priority Queue
  - Priority queues are a type of container adaptors, specifically designed such that its **first element is always the greatest** of the elements it contains, according to some strict weak ordering criterion.
- Member Function
  - push
  - pop
  - **top**
  - empty
  - size



made by electron & free999

(補充) Priority Queue

- **User-Defined Structure**

```
typedef structure _PRICE{
        int value;

        bool operator<(const structure _PRICE a) const {
                return value > a.value;
        }
} price;

int main()
{
        priority_queue<price> pq;
        price p1, p2;
        p1.value = 10;      p2.value = 5;
        pq.push(p1);        pq.push(p2);

        printf("top element's value = %d\n", pq.top().value);
        /* top element's value = 5 */
}
```

Practice 3

Uva - 11995

I Can Guess the Data Structure

made by electron & free999

Q&A
Thank for Your Attention

made by electron & free999