基礎班 STL(1)

By Chess

Outline

- Vector
- Iterator
- Stack
- Queue
- Deque
- Bitset

Vector

- 課金版的Array
- 可以隨時變動陣列長度

宣告

```
vector<int> first;                                  // empty vector of ints
vector<int> second(4, 100);                         // four ints with value 100
vector<int> third(second.begin(), second.end());    // iterating through second
vector<int> fourth(third);
```

Member function

- `empty()` // 檢查是否為空
- `size()` // 回傳vector中元素個數
- `erase(位置)` // 清除該元素
- `clear()` // 清空整條vector
- `capacity()` // 回傳vector容量
- `resize(值)` // 重定vector個數
- `reserve(值)` // 重定vector容量

Iterator

- STL容器中不可或缺的東西
- 功能強大 類似於pointer

- begin()　　// 指向容器的頭
- end()　　// 指向容器的尾"的下一個"

- Example:

```
vector<int> vec;

for(int i=0; i<10; i++)
    vec.push_back(i);

vector<int>::iterator it = vec.begin();

for(it; it!=vec.end(); it++)
    printf("%d\n", *it);
```

Stack

- 先進後出
- 疊盤子

Member function

- `empty()` // 檢查是否為空
- `size()` // 回傳stack中元素個數
- `push(值)` // 將元素推入stack中
- `pop()` // 將stack中最上面元素彈出來
- `top()` // 回傳目前最上面的元素

清空整個stack

```
stack<int> s;

while(!s.empty())
    s.pop();
```

練習

- UVa
  673

Queue

- 先進先出
- 排隊（台灣最美的風景是插隊(X
- 只能從屁股放

Member function

- empty() // 檢查是否為空
- size() // 回傳queue中元素個數
- push(值) // 將元素推入queue中*
- pop() // 將queue中最前面元素彈出來*
- front() // 回傳queue最前端元素
- back() // 回傳queue最尾端元素

Priority_Queue

- 與Queue幾乎相同，多了自動排序

宣告

```
priority_queue<int> pq;
```

Bonus

```
priority_queue<int, vector<int>, less<int> > pq;
                                    ^
                                   /|\
                                    |
                                    |

                                   or
                                greater<int>
```

Deque

- 大致上跟Queue是相同的
- 唯一的差別是Deque可以頭或從屁股放(雙向的)
- 有operator[ ]

Member function

- `push_front(值)` `// 將資料從最前面放進去`
- `push_back(值)` `// 從最後面放進去`
- `pop_front()` `// 彈出最前面的元素`
- `pop_back()` `// 彈出最後面的元素`

練習

- •UVa 11995
- •UVa 11997

Bitset

- 二進位相關處理

宣告

```
bitset<8> b;
bitset<8> b2(100ULL);
bitset<8> b3(077);
bitset<8> b4(0x0F);
bitset<8> b5(a);
```

運算

- NOT : ~
- OR : |
- AND : &
- XOR : ^
- >> : 右移
- << : 左移
- b[n] : 直接對第n bit做更改

Member function

- `flip()  // 將全部b的bit做not(=~b)`
- `reset() // 將全部bit設為0`
- `set()   // 將全部bit設為1`
- `count() // 回傳有幾個bit為1`
- `size()  // 回傳有幾個bit`

轉型

- to_string() // 將bit轉成string
  (還是以01儲存)
- to_ulong() // 將bit轉成unsigned long
  (變成10進位UL)
- to_ullong() // 將bit轉成unsigned long long
  (變成10進位ULL)

END