



SW 역량 테스트 대비반 - 2회차

2019년 04월 25일

지난 수업 정리

- 테스트 환경에 대한 적응 (<https://www.onlinegdb.com/>)
- 테스트 문제 풀이
- C++ 기본 입출력
- C++ STL : Stack, Queue, Vector 일부
- using namespace std;

Stack

라이브러리 include : `#include <stack>`
선언 : `stack <int> s;`

멤버 함수	기능
<code>s.size()</code>	s의 사이즈(물리적인 저장 용량이 아닌 원소의 개수)를 리턴
<code>s.empty()</code>	s의 사이즈가 0인지 아닌지를 확인
<code>s.top()</code>	s에 가장 나중에 들어간 원소를 리턴
<code>s.push(val)</code>	s의 뒤에 val 추가
<code>s.pop()</code>	s에 가장 나중에 들어간 원소를 삭제

Queue

라이브러리 include : `#include <queue>`
선언 : `queue <int> q;`

멤버 함수	기능
<code>q.size()</code>	q의 사이즈(물리적인 저장 용량이 아닌 원소의 개수)를 리턴
<code>q.empty()</code>	q의 사이즈가 0인지 아닌지를 확인
<code>q.front()</code>	q에 가장 먼저 들어간 원소를 리턴
<code>q.back()</code>	q에 가장 나중에 들어간 원소를 리턴
<code>q.push(val)</code>	q의 위(뒤에 val 추가
<code>q.pop()</code>	q에 가장 먼저 들어간 원소를 삭제

오늘 할 내용

구분	상세 내용	교육 날짜
1주차	운영계획 및 SW 역량 레벨 테스트	04/04
2주차	C/C++ 기초 문법 강의 및 실습 1	04/11
3주차	C/C++ 기초 문법 강의 및 실습 2	04/18
4주차	기초 자료구조 강의 및 실습	04/25
5주차	<div>vector(dequeue)</div> <div>list</div> <div>set, map, pair</div> <div>관련 문제 풀이</div>	05/09
6주차		05/16
7주차		05/23
8주차		05/30
9주차		06/07
10주차		06/13
11주차		06/20
12주차	알고리즘 성능 향상 기법	06/27
13주차 ~	실전 감각 기르기 (레벨에 따라 2~3개 분반 필요할 수 있음)	7월 부터는 주 2회 (화/목)

vector (deque)

정의 : 동적 배열

장점 : 배열 크기가 유동적, 데이터의 위치를 안다면 배열처럼 쉽게 접근 가능

단점 : 중간 값 삽입 삭제가 쉽지 않다. (shift되므로 무거움)데이터가

순차적으로 저장 되므로 검색속도가 빠르지 않음

<https://blockdmask.tistory.com/70>

<https://blockdmask.tistory.com/73>

적어도 이것만은 명확히 알자

- 벡터의 선언/초기화 방법
- 벡터의 처음과 끝, 삽입/삭제
- 벡터의 크기
- iterator

deque는 배열 앞에
삽입/삭제가 필요할
때 사용하자

list

정의 : 더블 링크드리스트

장점 : 포인터로 다음 값을 찾아주는 방식이므로 모든 삽입 삭제가 용이함.

단점 : vector에서 가능했던 데이터의 위치로 값에 접근할 수 없음.

<https://blockdmask.tistory.com/76>

적어도 이것만은 명확히 알자

- 리스트의 선언/초기화 방법
- 리스트의 처음과 끝, 삽입/삭제 (중간 포함)
- 리스트의 크기
- 정렬 방법

문제 풀이

<http://tech.kakao.com/2017/09/27/kakao-blind-recruitment-round-1/>

연산 순위	연산자	결합성
1	(), [], ->, .	좌→우
2	sizeof, &, ++, ==, ~, !, *(간접 지정 연산자), +(단항 연산자), -(단항 연산자)	좌←우
3	*(곱셈연산자), /(나눗셈연산자), %(나머지연산자)	좌→우
4	+(이항연산자), -(이항연산자)	좌→우
5	<<, >>	좌→우
6	<, <=, >=, >	좌→우
7	==, !=	좌→우
8	&	좌→우
9	^	좌→우
10		좌→우
11	&&	좌→우
12		좌→우
13	?:(삼항연산자)	좌←우
14	=, +=, *=, /=, %=, ^=, =, <<=, >>=	좌←우
15	,(coma연산자)	좌→우

1) <algorithm> 헤더를 include하여 reverse 함수를 사용한다.

```
1 #include <vector>
2 #include <algorithm>
3
4 int main() {
5     std::vector<int> a;
6     std::reverse(a.begin(), a.end());
7     return 0;
8 }
```

문제 풀이

답안

```
#include <iostream>
#include <vector>
#include <algorithm>
//#include <bits/stdc++.h>

using namespace std;
vector<int> arr1, arr2;
vector<string> ret;
vector<string> solution(int n, vector<int> a, vector<int> b)
{
    vector<string> res;
    for(int i=0; i<n; i++)
    {
        string st="";
        int tmp = a[i] | b[i];
        for(int j=0; j<n; j++)
        {
            if(tmp % 2 == 0) st += " ";
            else st += "#";
            tmp = tmp >> 1;
        }
        reverse(st.begin(), st.end());
        res.push_back(st);
    }
    return res;
}
```

```
int main()
{
    int n,x;
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&x);
        arr1.push_back(x);
    }
    for(int i=0; i<n; i++)
    {
        scanf("%d",&x);
        arr2.push_back(x);
    }
    ret = solution(n, arr1, arr2);
    for(int i=0; i<n; i++)
    {
        cout << ret[i] << endl;
    }
    return 0;
}
```

set

정의 : 연관 컨테이너 중 단순한 컨테이너로 key라 불리는 원소(value).의

집합으로 이루어진 컨테이너

key를 신속하게 찾고, 또 key가 정렬되기를 원할 때 사용됨

삽입 시 정렬이 이루어짐

<https://blockdmask.tistory.com/79>

적어도 이것만은 명확히 알자

- set의 선언/초기화 방법
- set의 처음과 끝 (s.begin(), s.end()), 삽입/삭제(s.insert(k), s.erase(iter))
- set의 원소 개수(s.size()), 검색(s.find(k))

map

정의 : 연관 컨테이너 중 자주 사용하는 컨테이너로 원소를 key 와 value의 쌍으로 저장

key를 신속하게 찾고, 또 key가 정렬되기를 원할 때 사용됨
삽입 시 정렬이 이루어짐

<https://blockdmask.tistory.com/87?category=249379>

적어도 이것만은 명확히 알자

- map의 선언/초기화 방법 (`map<int, int> m`)
- map의 처음과 끝 (`m.begin()`, `m.end()`), 삽입/삭제(`m.insert(pair [k,v])`, `m.erase(k)`)
- map의 원소 개수(`m.size()`), 검색(`m.find(k)`)

pair

정의 : 두 객체를 하나의 객체로 취급할 수 있게 묶어주는 클래스
key를 신속하게 찾고, 또 key가 정렬되기를 원할 때 사용됨
삽입 시 정렬이 이루어짐

<https://blockdmask.tistory.com/64?category=249379>

적어도 이것만은 명확히 알자

- pair의 선언/초기화 방법
- make_pair()
- vector, list 등과의 함께 사용하는 방법

문제 풀이

<https://www.acmicpc.net/problem/13414>

*힌트

- map, pair, vector, sort를 활용
- map의 특성상 이미 존재하는 key값을 갖는 pair를 map에 추가하는 경우, 해당 key값을 갖는 기존 pair의 value를 덮어 씌

문제 풀이

답안

```
#include <bits/stdc++.h>
```

```
using namespace std;
int K, L;
map<int,int> A;
map<int,int>::iterator it;
vector<pair<int,int> > B;
```

```
int main()
{
    scanf("%d%d", &K, &L);

    for(int i=1; i<=L; i++)
    {
        int s; scanf("%d", &s);
        A[s] = i;
    }
}
```

```
for(it=A.begin(); it!=A.end(); it++)
    B.push_back(make_pair(it->second, it->first));
```

```
//수강 신청 순서 기준으로 정렬
sort(B.begin(), B.end());
```

```
//수강 신청에 성공한 학생이 수강 가능 정원보다 적은 경우를
//고려해야 함
int num = min(K,int(B.size()));
```

```
for(int i=0; i<num; i++)
    printf("%d\n", B[i].second);
}
```

기타

1. 강의 자료 등록

- 네이버 클라우드(cloud.naver.com)에 공유
- 본인의 네이버 메일 주소를 적어주세요.

2. 숙제

- 제출 형식: 파일 이름은 본인 이름으로 제출해주세요.
- 제출 기한 : 다음 수업 전까지 (05.09 이전)
- 문제/제출 위치 :

https://cloud.naver.com/#WORKS=false&MSGN=ACTION_GET_LIST_111&MSGV=L1NX7Jet65-J7YWM7lqk7Yq464yA67mE67CYL-ylmeygnC8yMDE5MDQyNS8&PAGEV=1

(숙제-20190425 폴더)

- 다른 사람이 푼 거 참조하지 말고 스스로 고민해서 풀어보세요