SW 역량 테스트 대비반 – 2회차

2020년 07월 15일

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

Queue

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

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

커리큘럼

구분	내용
1주차 (07/08)	수업 방향 설명 및 C++ 기초 문법 강의 및 실습
2주차 (07/15)	기초 자료구조 강의 및 실습
3주차 (07/22)	재귀함수/탐색 알고리즘 강의 및 실습 1
4주차 (07/29)	
5주차 (08/05)	
6주차 (08/12)	
7주차 (08/19)	
8주차 (08/26)	
(옵션) 9주차 ~	

vector, list, set, map, pair,

algorithm 헤더

기타 헤더(math, cctype)

관련 문제 풀이

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 끝, 삽입/삭제**
- **벡터의 크기**
- **iterator**

deque는 배열 앞에 삽입/삭제가 필요할 때 사용하자

list

정의 : 더블 링크드 리스트

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

단점 : vector에서 가능했던 데이터의 위치로 값에 접근할 수 없음.

(순차접근)

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 1

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	,(콤마연산자)	좌→우

1. 비밀 지도(난이도: 하)

1) <algorithm> 헤더를 include하여 reverse 함수를 사용한다.

```cpp
#include <vector>
#include <algorithm>

int main() {
  std::vector<int> a;
  std::reverse(a.begin(), a.end());
  return 0;
}
```

문제 풀이 1

아이디어

1. 2개의 int 벡터에 각각 n개의 정수를 입력 받는다.
2. n까지 루프를 각 벡터의 값을 bit 연산한다.
   - %2로 0이면 공백, 1이면 #을 넣어준다.
   - reverse 메소드 수행 후 리턴 벡터에 넣어준다.
3. 위의 2을 입력 받은 수만큼(n번) 수행한다.

문제 풀이 1

답안

https://onlinegdb.com/B1nQPAskw

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

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 2

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 씀**

문제 풀이 2

아이디어

1. **map의 특성에 맞게(key는 중복 허용이 되지 않음, overwrite)**

   학번을 key로 하고 입력 순서를 value로 하는 map으로 입력을 받는다.

2. **정렬을 위해 vector를 map의 (value, key) pair로 입력한다.**

3. **pair의 value 기준으로 정렬을 한다.**

4. **min(vector의 개수, 수강가능인원) 만큼 출력한다.**

문제 풀이 2

답안

https://onlinegdb.com/ryuUiCs1D

중간 정리

- **vector(deque)**
  - **동적 배열, 선언, 초기화, 삽입/삭제, iterator 활용**
- **list**
  - **더블 링크드 리스트, 선언, 초기화, 삽입/삭제**
- **set : key로 구성된 컨테이너, key로 정렬됨**
- **map : (key, value)로 구성된 컨테이너, key로 정렬됨**
- **pair : 두 객체를 하나의 객체로 취급할 수 있게 묶어주는 클래스**

간단하게 쓰고 싶을 때는 배열~
동적으로 쓰고 싶다면 vector 나 list~
여기서 번호(key)로 찾고 싶으면(랜덤검색) vector를 쓰고 중간에 값을 넣거나 빼고 싶으면 list를 써라
만약 검색을 자주 할거 같으면 map을 쓸것이고(정렬되있음)(삽입/삭제 자주 하지 말 것)
key만 필요하면 set을 써라(정렬되있음)
구조체 형태로 쓸 꺼면 pair를 활용

algorithm 헤더 – Sorting

Sorting:

sort	Sort elements in range (function template)
stable_sort	Sort elements preserving order of equivalents (function template)
partial_sort	Partially sort elements in range (function template)
partial_sort_copy	Copy and partially sort range (function template)
is_sorted C++11	Check whether range is sorted (function template)
is_sorted_until C++11	Find first unsorted element in range (function template)
nth_element	Sort element in range (function template)

- **sort : https://onlinegdb.com/Bkv5w612E**
- **partial_sort : https://onlinegdb.com/ryjJqaknE**
- **is_sorted : https://onlinegdb.com/HkRNhp12N**

algorithm 헤더 – Non-modifying sequence operations

find	Find value in range (function template)
find_if	Find element in range (function template)
find_if_not C++11	Find element in range (negative condition) (function template)
find_end	Find last subsequence in range (function template)
find_first_of	Find element from set in range (function template)
adjacent_find	Find equal adjacent elements in range (function template)
count	Count appearances of value in range (function template)
count_if	Return number of elements in range satisfying condition (function template)
mismatch	Return first position where two ranges differ (function template)
equal	Test whether the elements in two ranges are equal (function template)
is_permutation C++11	Test whether range is permutation of another (function template)
search	Search range for subsequence (function template)
search_n	Search range for elements (function template)

- **find, find_if : https://onlinegdb.com/Sk13h01nN**
- **count, count_if : https://onlinegdb.com/HyxCaCJhE**
- **equal : https://onlinegdb.com/rk5Ga1xh4**
- **search : https://onlinegdb.com/HJV_lxe3E**

algorithm 헤더 – Modifying sequence operations

copy	Copy range of elements (function template)
copy_n C++11	Copy elements (function template)
copy_if C++11	Copy certain elements of range (function template)
copy_backward	Copy range of elements backward (function template)
move C++11	Move range of elements (function template)
move_backward C++11	Move range of elements backward (function template)
swap	Exchange values of two objects (function template)

remove	Remove value from range (function template)
remove_if	Remove elements from range (function template)

transform	Transform range (function template)

- copy, copy_if : https://onlinegdb.com/rJFYExgnN
- swap : https://onlinegdb.com/HkJcIlx2N
- remove, remove_if : https://onlinegdb.com/rJHWcxe34
- transform : https://onlinegdb.com/HJNLjfxnV

algorithm 헤더 – Min/Max

Min/max:

min	Return the smallest (function template)
max	Return the largest (function template)
minmax C++11	Return smallest and largest elements (function template)
min_element	Return smallest element in range (function template)
max_element	Return largest element in range (function template)
minmax_element C++11	Return smallest and largest elements in range (function template)

- **https://onlinegdb.com/B1yinbe3V**

algorithm 헤더

<Reference>

http://www.cplusplus.com/reference/algorithm/

https://modoocode.com/225

https://modoocode.com/256

문제 풀이 3

문제

알파벳 소문자로 이루어진 N개의 단어가 들어오면 아래와 같은 조건에 따라 정렬하는 프로그램을 작성하시오.

1. 길이가 짧은 것부터
2. 길이가 같으면 사전 순으로

입력

첫째 줄에 단어의 개수 N이 주어진다. (1≤N≤20,000) 둘째 줄부터 N개의 줄에 걸쳐 알파벳 소문자로 이루어진 단어가 한 줄에 하나씩 주어진다. 주어지는 문자열의 길이는 50을 넘지 않는다.

출력

조건에 따라 정렬하여 단어들을 출력한다. 단, 같은 단어가 여러 번 입력된 경우에는 한 번씩만 출력한다.

예제 입력 1 복사

```
13
but
i
wont
hesitate
no
more
no
more
it
cannot
wait
im
yours
```

예제 출력 1 복사

```
i
im
it
no
but
more
wait
wont
yours
cannot
hesitate
```

문제 풀이 3

아이디어

1. **입력 vector에 <단어,단어의 길이> 를 넣어준다.**
2. **소팅해준다.**
   - **길이가 작은 순으로..**
   - **길이가 같을때 문자끼리 비교해서 작은게 앞으로 오게한다.**
3. **중복 처리하며 출력한다.**

문제풀이 3

답안

https://onlinegdb.com/HkTIUq9YH

문제 풀이 4

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

5. 뉴스 클러스터링(난이도: 중)

여러 언론사에서 쏟아지는 뉴스, 특히 속보성 뉴스를 보면 비슷비슷한 제목의 기사가 많아 정작 필요한 기사를 찾기가 어렵다. Daum 뉴스의 개발 업무를 맡게 된 신입사원 튜브는 사용자들이 편리하게 다양한 뉴스를 찾아볼 수 있도록 문제점을 개선하는 업무를 맡게 되었다.

개발의 방향을 잡기 위해 튜브는 우선 최근 화제가 되고 있는 “카카오 신입 개발자 공채” 관련 기사를 검색해보았다.

- 카카오 첫 공채..’블라인드’ 방식 채용
- 카카오, 합병 후 첫 공채.. 블라인드 전형으로 개발자 채용
- 카카오, 블라인드 전형으로 신입 개발자 공채
- 카카오 공채, 신입 개발자 코딩 능력만 본다
- 카카오, 신입 공채.. “코딩 실력만 본다”
- 카카오 “코딩 능력만으로 2018 신입 개발자 뽑는다”

기사의 제목을 기준으로 “블라인드 전형”에 주목하는 기사와 “코딩 테스트”에 주목하는 기사로 나뉘는 걸 발견했다. 튜브는 이들을 각각 묶어서 보여주면 카카오 공채 관련 기사를 찾아보는 사용자에게 유용할 듯싶었다.

문제 풀이 4

아이디어

1. 두개의 스트링을 입력받아 모두 소문자로 변환한다. (transform)
2. 첫번째 입력 스트링의 2개 문자 조합을 sum 한다. (map을 활용하며 중복 허용)
3. 두번째 입력 스트링의 2개의 문자 조합이 첫번째 문자 조합에 존재하면 교집합에 넣어주고, 존재하지 않으면 합집합에 넣어준다.
4. 조건에 맞게 자카드 유사도를 계산한다.

문제풀이 4

답안

https://onlinegdb.com/Bkomhfeh4

기타 헤더 – cmath

abs	Compute absolute value (function)

ceil	Round up value (function)
floor	Round down value (function)
round C++11	Round to nearest (function)

- **https://onlinegdb.com/ByEDDmxnV**

참고 :
http://www.cplusplus.com/reference/cmath/

기타 헤더 – cctype

isalnum	Check if character is alphanumeric (function)
isalpha	Check if character is alphabetic (function)
isblank C++11	Check if character is blank (function)
iscntrl	Check if character is a control character (function)
isdigit	Check if character is decimal digit (function)
isgraph	Check if character has graphical representation (function)
islower	Check if character is lowercase letter (function)
isprint	Check if character is printable (function)
ispunct	Check if character is a punctuation character (function)
isspace	Check if character is a white-space (function)
isupper	Check if character is uppercase letter (function)
isxdigit	Check if character is hexadecimal digit (function)

Character conversion functions

Two functions that convert between letter cases:

tolower	Convert uppercase letter to lowercase (function)
toupper	Convert lowercase letter to uppercase (function)

- **https://onlinegdb.com/rJ8NF7l2N**

참고 :
http://www.cplusplus.com/reference/cctype/?kw=cctype

문제 풀이 5

https://www.acmicpc.net/problem/14655

문제

욱제는 라스베이거스에서 유명한 베팅꾼이다. 어찌나 게임을 잘 하는지 '제2의 홍진호'라는 별명이 붙었을 정도다. 어찌나 게임을 잘 하는지 '제2의 홍진호'라는 별명이 붙었을 정도다.

욱제가 주로 하는 게임은 아주 단순하고, 친숙한 게임이다. 바로 동전 뒤집기 게임이다. 이 게임에 쓰이는 동전의 양면에는 절댓값이 같고 부호가 다른 정수가 한 면에 하나씩 쓰여 있다. (단, 동전끼리는 쓰인 숫자의 절댓값이 다를 수 있다) 한 플레이어 당 두 번의 라운드가 주어진다. 모든 라운드는 같은 동전으로 진행되며, 딜러는 각 라운드마다 N개의 동전을 임의로 섞고 이를 일렬로 배열한다. 이때, 동전의 앞뒤 방향도 바뀔 수 있다. 첫 번째 라운드에서는 동전에 표시된 값들의 합이 최대가 되도록 뒤집어야 하고, 두 번째 라운드에서는 동전에 표시된 값들의 합이 최소가 되도록 뒤집어야 한다. (첫 번째 라운드 동전 값의 합) - (두 번째 라운드 동전 값의 합)이 해당 플레이어가 게임에서 획득한 점수이고, 이 점수가 최대가 되는 플레이어가 바로 게임의 승자가 된다.

욱제는 엄지, 검지, 중지를 이용해서 항상 연속한 3개의 동전을 뒤집는 최고의 동전 뒤집러이다. 욱제는 연속한 3개의 동전을 뒤집지 않으면 이길 수 없다고 생각하기 때문에 실패하는 경우 없이 항상 연속한 3개의 동전만 뒤집는다. 동전 배열의 양 끝에서 벗어나서 양 끝의 동전만 뒤집거나 양 끝의 두 개 동전만 뒤집는 것도 가능하다. 동전을 뒤집는 횟수에 제한은 없다.

(!) 너, 강해 보이는군. 나와 승부를 겨루자! 띠리링띠리링디리 | 리리 | 링~ 앗! 심술쟁이 해커 임준오(동탄 주민)이 승부를 걸어왔다!

욱제는 이번 게임에서 얼마의 점수를 획득하게 될까? 욱제는 최고의 베팅꾼이기 때문에 항상 게임에서 획득할 수 있는 최고의 점수를 얻는다는 사실은 자명하다.

문제 풀이 5

아이디어

1. 문제의 아래 조건을 이해하는 것이 핵심인 문제였습니다.

- "욱제는 연속한 3개의 동전을 뒤집지 않으면 이길 수 없다고 생각하기 때문에 실패하는 경우 없이 항상 연속한 3개의 동전만 뒤집는다. 동전 배열의 양 끝에서 벗어나서 양 끝의 동전만 뒤집거나 양 끝의 두 개 동전만 뒤집는 것도 가능하다. 동전을 뒤집는 횟수에 제한은 없다."

2. 결국, 같은 동전을 1번 이상 뒤집을 수 있기 때문에 원하는 대로 동전을 배치할 수 있습니다.

3. 따라서 아래와 같이 그리디(Greedy)하게 접근할 수 있습니다.

4. 1 라운드에서는 합이 최대가 되도록 모두 양수로 만들어줍니다.

5. 2 라운드에서는 합이 최소가 되도록 모두 음수로 만들어줍니다.

6. 따라서 (1 라운드 - 2 라운드)는 모든 동전에 적힌 숫자의 절대값들의 합입니다!

문제풀이 5

답안

https://onlinegdb.com/S1glqyGZ2N

문제 풀이 6

http://tech.kakao.com/2017/11/14/kakao-blind-recruitment-round-3/

문제2. 압축

신입사원 어피치는 카카오톡으로 전송되는 메시지를 압축하여 전송 효율을 높이는 업무를 맡게 되었다. 메시지를 압축하더라도 전달되는 정보가 바뀌어서는 안 되므로, 압축 전의 정보를 완벽하게 복원 가능한 무손실 압축 알고리즘을 구현하기로 했다.

어피치는 여러 압축 알고리즘 중에서 성능이 좋고 구현이 간단한 **LZW**(Lempel–Ziv–Welch) 압축을 구현하기로 했다. LZW 압축은 1983년 발표된 알고리즘으로, 이미지 파일 포맷인 GIF 등 다양한 응용에서 사용되었다.

LZW 압축은 다음 과정을 거친다.

1. 길이가 1인 모든 단어를 포함하도록 사전을 초기화한다.
2. 사전에서 현재 입력과 일치하는 가장 긴 문자열 `w` 를 찾는다.
3. `w` 에 해당하는 사전의 색인 번호를 출력하고, 입력에서 `w` 를 제거한다.
4. 입력에서 처리되지 않은 다음 글자가 남아있다면(`c`), `w+c` 에 해당하는 단어를 사전에 등록한다.
5. 단계 2로 돌아간다.

압축 알고리즘이 영문 대문자만 처리한다고 할 때, 사전은 다음과 같이 초기화된다. 사전의 색인 번호는 정수값으로 주어지며, 1부터 시작한다고 하자.

문제 풀이 6

아이디어

1. 사전을 이용하라는 말에 바로 map이라는 자료구조를 이용 (dict)
2. 영문 대문자 색인 초기화 (1 ~ 26)
3. 가장 긴 문자열 w를 찾기 위해서는 글자를 붙여가며 사전에 있는지 찾아보다가 사전에 존재하지 않는다면 그 전의 문자열이 가장 긴 w이기 때문에 해당 문자열의 색인번호를 answer에 추가해주고 해당하는 새로운 문자열은 색인에 추가
4. 가장 긴 문자열을 찾으려할 때 만약 msg의 가장 마지막 문자열을 붙였다면 그 문자열까지가 가장 긴 문자열 w에 해당하므로 색인 번호를 answer에 추가

문제풀이 6

답안

https://onlinegdb.com/H1i-74Z2E