

2020년 ALPS 여름방학 비대면 스터디

- 4주차 -
문제풀이

참여방법

- 슬랙 - 코드 공유용으로 사용
- 구글 meet - 화면공유 및 음성으로 코드 설명 및 서로간 피드백 (<https://meet.google.com/gpu-mixd-bqa>) 링크는 계속 고정입니다.

항상 위 링크로 들어오시면 됩니다.

슬랙과 meet를 같이 띄우고 진행해주시면 감사하겠습니다.

참여방법 / 슬랙



Join your team on Slack

has invited you to use Slack with them,
in a workspace called **ALPS Study 2020.07**.



ALPS Study 2020.07

alps-study-202007.slack.com

Join Now

has already joined



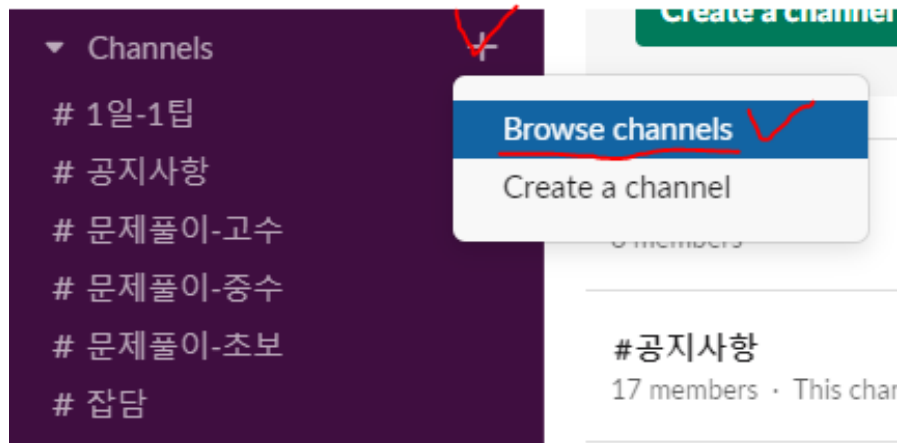
메일로 보내드린
초대장에 Join Now를
클릭하시면
입장 가능합니다

참여방법/구글 meet

- <https://meet.google.com/gpu-mixd-bqa>
- 해당 링크로 접속
- 해당 스터디 진행영상은 참여하지 못하신 분들을 위해 녹화 후 유튜브 채널에 업로드할 예정입니다^^
- 혹시 녹화를 원하지 않는 분들은 말씀해주세요.
- 아래는 영상이 올라갈 주소 입니다.
- <https://www.youtube.com/playlist?list=PL9gVcwpebJSJJ80vNpdrYAhyG6PXxbgq>

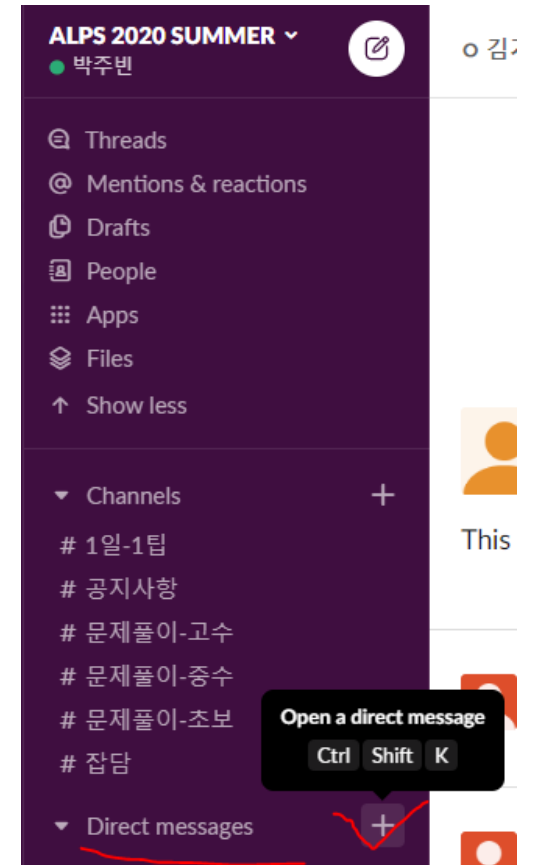
#1일-1팁

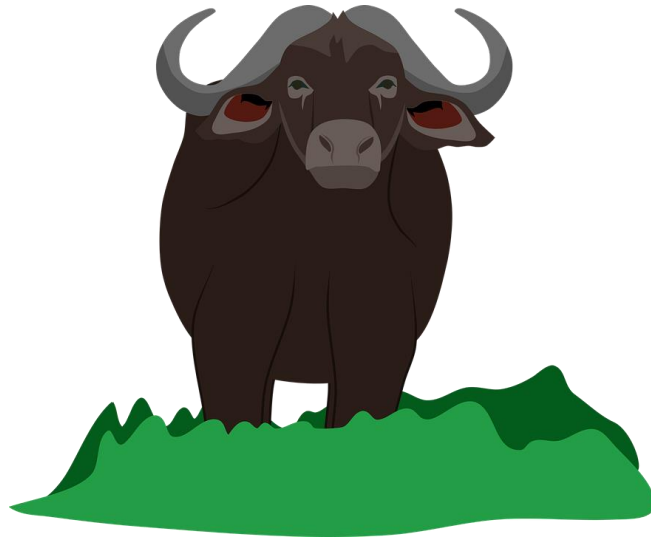
- 슬랙방에 1일-1팁 채널을 생성했습니다
- 하루에 한 번씩 코딩에 관한 꿀팁을 제공하니 많이 얻어가요



시작하기 전에..

- 이번 4주차에는 먼저 **55분** 동안 모두에게 문제 푸는 시간이 주어집니다.
 - 도중에 궁금한 문제나 풀리지 않는 문제가 있다면 **슬랙방**에 질문을 하시고, **익명**으로도 가능하니 스터디 운영진에게 **Direct Message**를 통해 알려주세요.
- 나머지 **55분**은 풀이 설명을 진행할 것입니다.
 - 시청은 **자유**이며, 여러분이 **해당 문제**의 설명을 듣지 않아도 된다고 판단하면 그 시간에 다른 문제를 푸셔도 좋습니다!





<https://pixabay.com/vectors/african-buffalo-buffalo-africa-5394717/>

완전탐색, Brute-Force Search

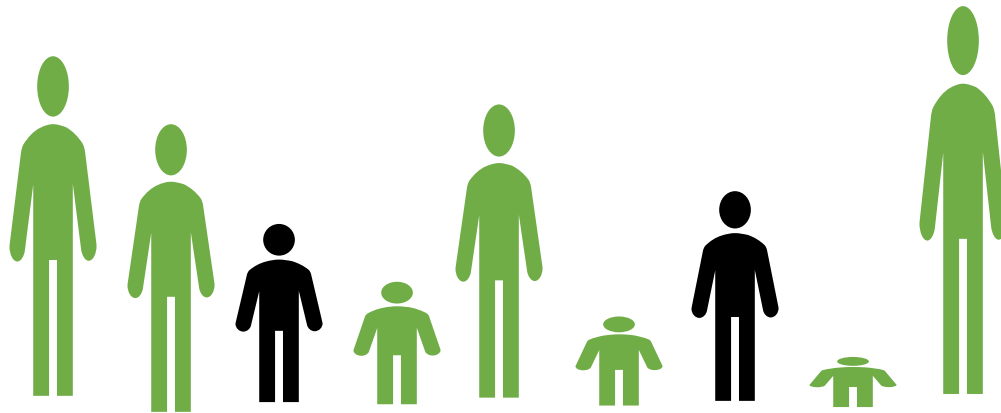
모든 경우의 수를 다 고려해본다.
단순무식 그러나 강력하고 정확함
Exhaustive Search

boj-2309 일곱 난쟁이 (난이도 : 하)

<https://www.acmicpc.net/problem/2309>

boj-2309 일곱난쟁이

- 9명 중 7명의 키의 합이 100이 되어야 함
- 주먹구구식 접근
- 7중 for문 사용
 - 그러기에는 하드코딩을 해야함
 - 더 좋은 방법이 있을까?



boj-2309 일곱난쟁이

- 발상의 전환
 - 9명의 키를 모두 더하고 그 중 2명의 키를 빼서 100이 되게 한다면?
 - 이전 방법보다 더 효율적
- 조합 $C(9,7) = C(9,2)$
- 2중 for문으로도 해결 가능!



boj-2309 일곱난쟁이

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week4/boj-2309.cpp
- <https://kjwsx23.tistory.com/355> (백트래킹)

programmers-42841 숫자 야구 (난이도 : 중)

<https://programmers.co.kr/learn/courses/30/lessons/42841>

programmers-42841 숫자 야구

- 정답 숫자와 내가 만든 숫자를 비교해서
- 스트라이크/볼 결과가 모두 같으면 정답의 후보

programmers-42841 숫자 야구

- 게임은 여러 턴을 거친다.
 - 정답 후보가 다음 턴에서는 후보가 안될 수 있음
- 거꾸로 생각하기
- 여집합?
 - 처음부터 모든 숫자가 정답의 후보로 지정
 - 턴마다 정답과 모든 숫자를 비교해서 결과가 **다른 경우에** 후보 명단에서 지운다!

programmers-42841 숫자 야구

- 규칙: 1~9 사이 숫자를 한 번씩만 사용해서 3자리의 숫자를 완성
 - $9 \times 8 \times 7 = 504$ 개의 숫자
- 숫자들은 어떻게 구할까요?
 1. 123~987 사이의 숫자를 모두 비교
 - 규칙에 맞지 않은 $(987 - 123) - 504 = 360$ 개의 숫자를 더 비교해야 함
 2. 123~987 사이의 숫자 중 가능한 경우만 추출해서 배열에 미리 저장
 - 모든 경우를 따지긴 해야 함. 하지만 최초 한 번만 실행
- 어떤 방법을 쓸 것인지는 자유
 - 숫자가 그렇게 크지 않기 때문
 - 발표자는 2번 방법을 사용

programmers-42841 숫자 야구

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week4/programmers-42841.cpp

boj-3085 사탕 게임

(난이도 : 중)

<https://www.acmicpc.net/problem/3085>

boj-3085 사탕 게임

- 문제가 어렵게 보일 수 있지만 차근차근 생각해보면 해결 가능

1. 행 위주로 비교

- 각 **행**마다 인접한 사탕을 swap, 연속된 사탕 개수 취득, 다시 swap

2. 열 위주로 비교

- 각 **열**마다 인접한 사탕을 swap, 연속된 사탕 개수 취득, 다시 swap

- 말은 쉽다!!

boj-3085 사탕 게임

- swap를 자주 사용할 것 같으니 함수로 만든다!
 - 행의 길이를 자주 구하는 것 같으니 함수로 만든다!
 - 열의 길이를 자주 구하는 것 같으니 함수로 만든다!
-
- 이렇게 코드 중복이 된다 싶을 때 함수로 만들면 편안해져요!



boj-3085 사탕 게임

- 다 같이 생각해 봅시다

?	?	?	?
?	a	b	?
?	?	?	?
?	?	?	?

?	?	?	?
?	?	b	?
?	?	c	?
?	?	?	?

boj-3085 사탕 게임

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week4/boj-3085.cpp

programmers-42840 모의고사 (난이도 : 하)

<https://programmers.co.kr/learn/courses/30/lessons/42840>

programmers-42840 모의고사

- 1,2,3번 수포자는 항상 순서대로 같은 번호를 찍는다.
- 따라서 해당 번호를 배열에 미리 넣어두고 주어진 `vector<int> answers`를 완전탐색한다.

programmers-42840 모의고사

- `vector<int> answers`를 차례대로 탐색하면서 1,2,3번 수포자가 정답을 맞춘 횟수를 `cnt[i]`에 추가해준다.
- 모든 `cnt[i]`를 확인한 후엔 가장 많이 맞춘 사람들을 출력해야하는데

이 경우는 `cnt[1], cnt[2], cnt[3]`순서대로 탐색하면서

만약 `max < cnt[i]` 이면 `max`와 `answer`를 초기화 해주고

만약 `max == cnt[i]` 이면 `answer`에 추가해준다.(이전 사람과 맞힌 횟수가 같으므로)

programmers-42840 모의고사

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/moigosa.cpp

boj-14500 테트로미노 (난이도 : 상)

<https://www.acmicpc.net/problem/14500>

boj-14500 테트로미노

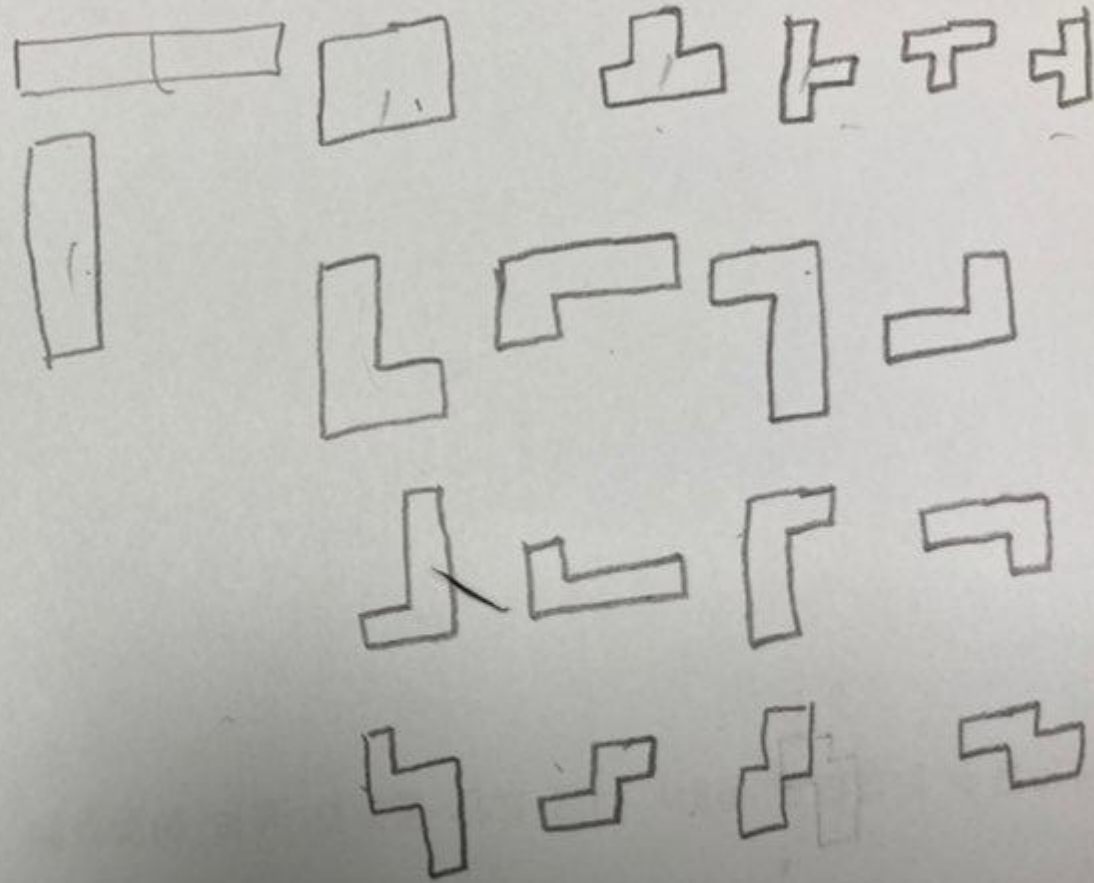
- 브루트포스(완전탐색) 혹은 dfs로 구할 수 있다.
- 브루트포스를 사용할 경우 모든 가능한 경우의 수를 구한다.

boj-14500 테트로미노

- 모든 블록의 경우의 수 대해서 좌표를 연산해 주어야한다.
이는 사전에 미리 계산해서 배열에 저장해주어야한다.
- 그리고 모든 좌표에 대해서 각각 블록을 배치해보고 $ans = \max(ans, \text{값})$ 해주어 정답을 구한다.
- 블록의 좌표를 미리 그리고 배열에 저장하는 과정에서 실수를 할 가능성이 높다.

모든 블록 이동 좌표 저장

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1		
3,0			



```
int block[4][2] = { {{0,0},{0,1},{0,2},{0,3}}, {{0,0},{1,0},{2,0},{3,0}}//작대기
,{{0,0},{0,1},{1,0},{1,1}}//뿔
,{{1,0},{1,1},{0,1},{1,2}},{{0,0},{1,0},{2,0},{1,1}},{{0,0},{0,1},{0,2},{1,1}}
,{{0,0},{1,0},{2,0},{2,1}},{{0,0},{1,0},{0,1},{0,2}},{{0,0},{0,1},{1,1},{2,1}}
,{{2,0},{2,1},{1,1},{0,1}},{{0,0},{1,0},{1,1},{1,2}},{{0,0},{0,1},{1,0},{2,0}}
,{{0,0},{1,0},{1,1},{2,1}},{{1,0},{1,1},{0,1},{0,2}},{{0,1},{1,0},{1,1},{2,0}}
};
```

boj-14500 테트로미노

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/14500.cpp
- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/14500_dfs.cpp(dfs로 풀 경우)

boj-1062 가르침 (난이도 : 상)

<https://www.acmicpc.net/problem/1062>

boj-1062 가르침

- 완전탐색(브루트포스)를 통해 모든 배울 수 있는 철자의 조합에 대해서 배울 수 있는 단어의 갯수를 찾는다.
- 조합의 갯수를 구하기 위해서 함수의 재귀호출 혹은 `next_permutation()` 함수를 사용한다.
- 재귀호출을 통해 조합을 구할 경우 이전 경우에 대한 기록이 필요
- `next_permutation(배열 시작 주소, 배열 끝 주소);`을 사용하면 간단히 순열 혹은 조합을 탐색할 수 있다.

next_permutation(v.begin(), v.end())

- 보통 do while문 형태로 쓰인다.
- 주어진 배열(벡터)의 숫자들에 대해서 하나의 순열로 보고 다음 순열을 해당 배열(벡터)에 업데이트해준다.
- 순열을 구할 경우
- 12345-> 12354 -> 12435 -> ... -> 54312 -> 54321
- 조합을 구할 경우
- 00111 -> 01011 -> 01101 -> ... -> 11100
- 순열과 조합 모두 각각의 모든 경우의 수를 구할 수 있다.

next_permutation(v.begin(), v.end())

```
do{  
    for(int i=0;i<K;i++){  
        cout << permu[i];  
    }  
    cout << '\n';  
    //ans = max(ans,countWord(permu));  
}while(next_permutation(permu.begin(),permu.end()));
```

12345
12354
12435
12453
12534
12543

boj-1062 가르침

- `next_permutation()`으로 모든 배울 수 있는 철자의 경우의 수를 찾고 각각의 경우마다 배울 수 있는 단어의 수를 구하고 `ans = max(ans, 값)`을 통해 정답을 업데이트해준다.
- $26C13$ 일 경우 많은 경우의 수를 연산하므로 `acnti`가 속하지 않는 경우는 배울 수 있는 단어의 수를 세지 않는다.

boj-1062 가르침

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/1062.c
[pp](#)

다음 5주차에는..

- 이분 탐색을 배웁니다.
- 감사합니다.