

2020년 ALPS 여름방학 비대면 스터디

- 5주차 -
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



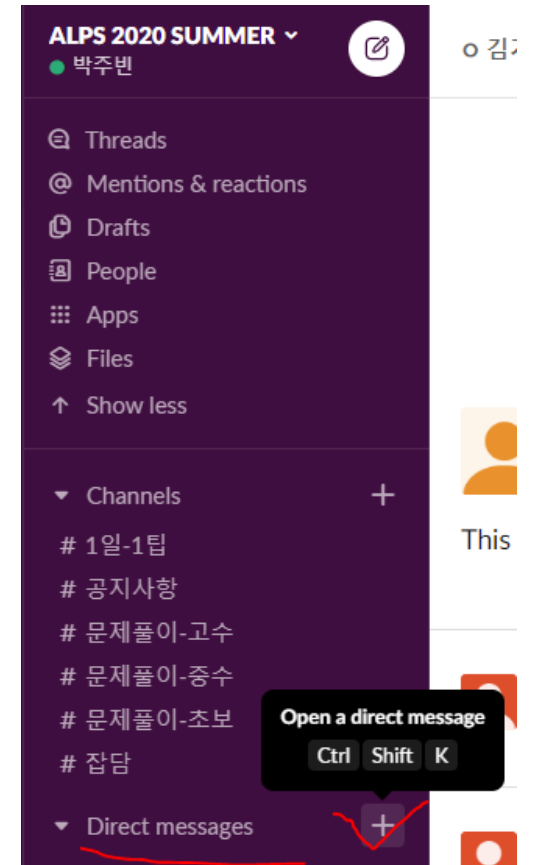
메일로 보내드린
초대장에 Join Now를
클릭하시면
입장 가능합니다

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>

시작하기 전에..

- 이번 5주차에는 먼저 **15분간** 개념 설명 뒤에
- **60분** 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 **40분**은 풀이 설명을 진행할 것입니다.
 - 시청은 **자유**이며, 여러분이 **해당 문제**의 설명을 듣지 않아도 된다고 판단하면 그 시간에 다른 문제를 푸셔도 좋습니다!



이분탐색

Binary Search

파라메트릭 탐색

Parametric Search

<https://sarah950716.tistory.com/16>

<https://velog.io/@funhan/%EC%9D%B4%EB%B6%84-%ED%83%90%EC%83%89-binary-search>

<https://ialy1595.github.io/post/parametric-search/>

공통점과 차이점

이분탐색

- 정렬된 상태여야 함
- 구간을 **반절**로 나누어 탐색
 - 내가 찾고자 하는 **답**을 구함
 - 그 답이 연속된 데이터에 들어 있는지 확인 수단으로밖에 쓰이지 않음

파라메트릭 탐색

- 정렬된 상태여야 함
- 구간을 **반절**로 나누어 탐색
 - 내가 찾고자 하는 답의, 가능한 **범위**를 구함
- 범위를 구한다는 것은?
 - ***min*, *MAX***를 구한다는 얘기
 - 이 문제를 **최적화 문제**라고 함
- 어떤 조건을 적용해 **최적화 문제**를 결정 문제(참/거짓)로 바꾸어 푸는 알고리즘

파라메트릭 탐색 조건

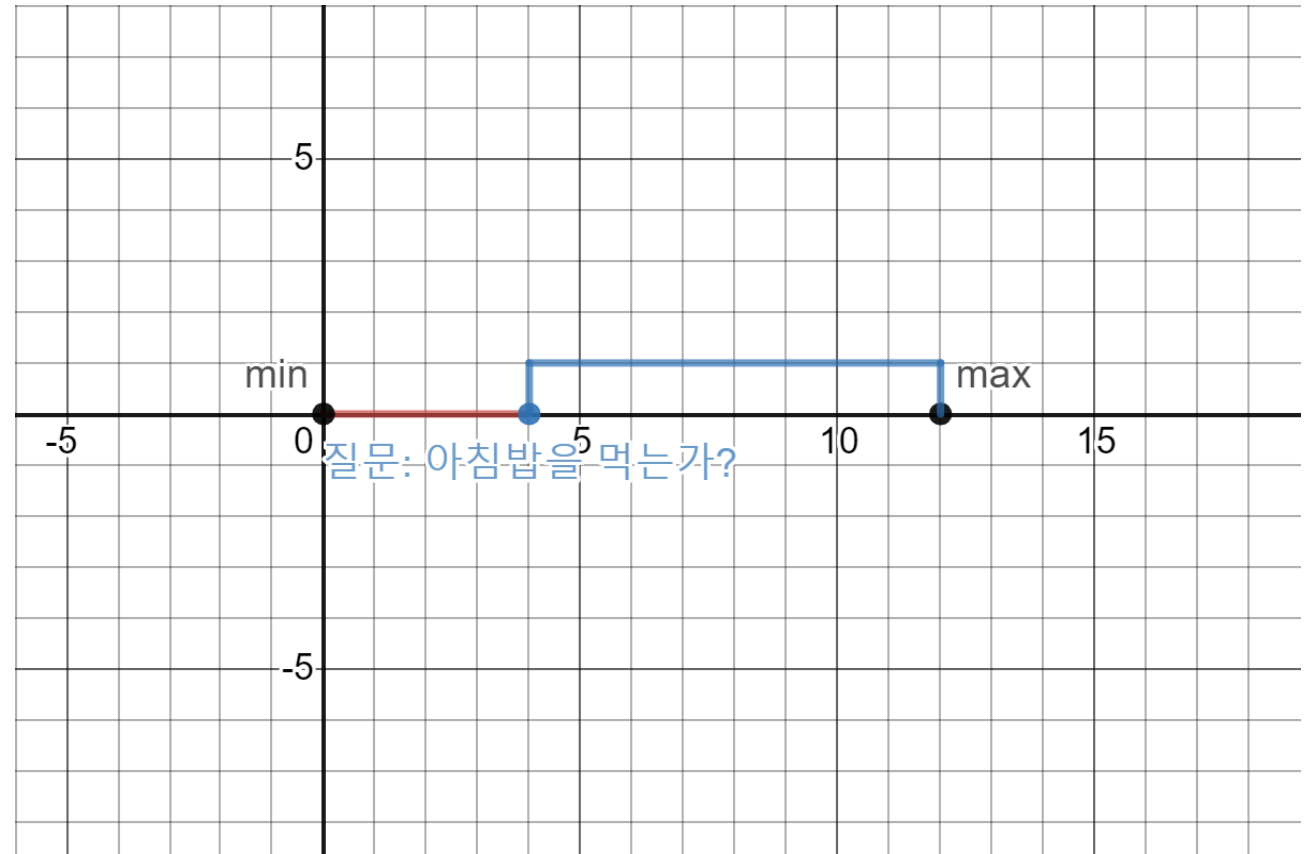
- 최대값 M 을 구하는 문제의 경우, M 이 어떤 조건 c 을 만족하면 M 보다 작은 값도 모두 조건 c 을 만족해야 함
- 최소값 m 을 구하는 문제의 경우, m 이 어떤 조건 c 을 만족하면 m 보다 큰 값도 모두 조건 c 을 만족해야 함
- 연속적

구간 표기법

- $a \leq x < b \iff [a, b)$
- $a \leq x \leq b \iff [a, b]$
- $a < x \leq b \iff (a, b]$
- $a < x < b \iff (a, b)$

파라메트릭 탐색

- 사람 13명에 대해 키를 오름차순으로 정렬
- **키가 166cm 이상**은 반드시 아침밥을 먹는다는 사실이 증명
- 질문(조건): 아침밥 드세요?
- 아침밥 안 먹으면 **0**
- 아침밥 먹으면 **1**
- 아침밥을 **먹는 사람** 중 가장 키가 작은 사람 구하기



파라메트릭 탐색 추적하기



시간 초과 주의!

- while문에서 left와 right를 비교할 때 시간초과에 걸리는 일이 빈번히 발생할 수 있음
- 둘 중에 하나를 사용하면 됩니다
- **[한 쪽 만족, 한 쪽 불만족]**
 - while (low + 1 < high), low = mid, high = mid
- **[양 쪽 만족]**
 - while (low <= high), low = mid + 1, high = mid - 1

boj-2792 보석상자 (난이도 : 중)

<https://www.acmicpc.net/problem/2792>

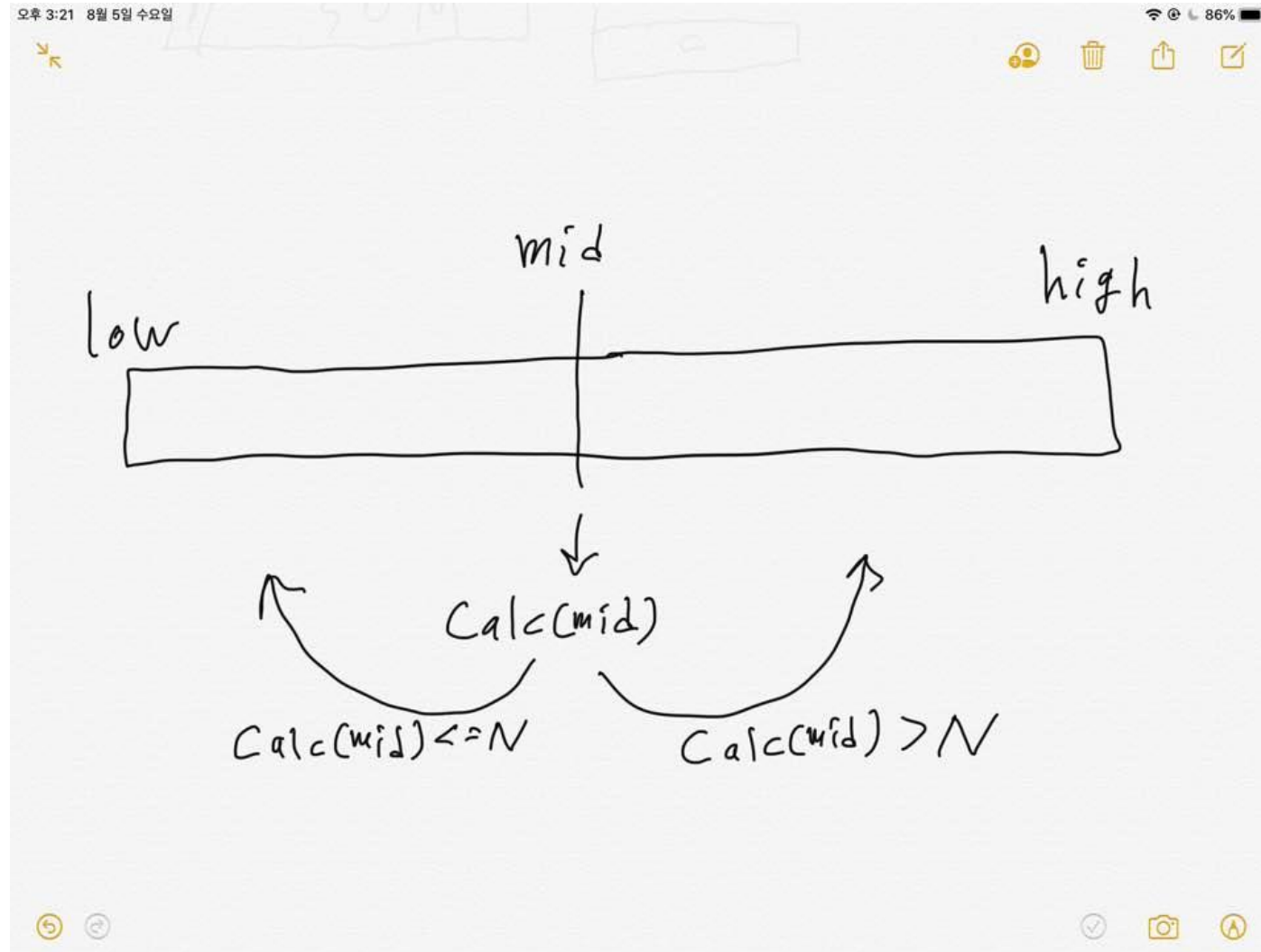
boj-2792 보석상자

- 보석을 몇개씩 나누어주어야 가장 적게 나누어 줄 수 있는가?
- 가장 적게 나누어 줄 수 있는 수는 이분탐색으로 구한다.
- 매번 나누어 줄 갯수에 대해서 최대 몇 명을 나누어 줄 수 있는지 연산해야하므로 확인하는 과정이 가장 적은 이분탐색이 적절하다.

boj-2792 보석상자

- If($\text{calc}(\text{mid}) \leq N$)에 대해서 mid개 만큼 학생들에게 나누어 준다고 할때 N명보다 적거나 같게 나누어 준다면 mid보다 작거나 같게 나누어 주어야한다.
- else에 대해서 N명보다 크게 나누어 주게 된다면 mid개 보다 크게 나누어 주어야한다.
- $\text{calc}(\text{mid})$ 함수는 mid갯수 만큼 나누어 주었을때 총 나누어 줄 수 있는 학생수를 반환합니다.

boj-2792 보석상자



boj-2792 보석상자

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/2792.cpp
- <https://jaimemin.tistory.com/1128>

boj-2512 예산 (난이도 : 중)

<https://www.acmicpc.net/problem/2512>

boj-2512 예산

- 적절한 예산 최대값(maximum)을 찾기 위해서 이분탐색을 사용
- 이때 이분탐색내에 비교조건값은 최종 예산값합이 M보다 큰지 작은지를 비교하는 것이 좋습니다.

boj-2512 예산

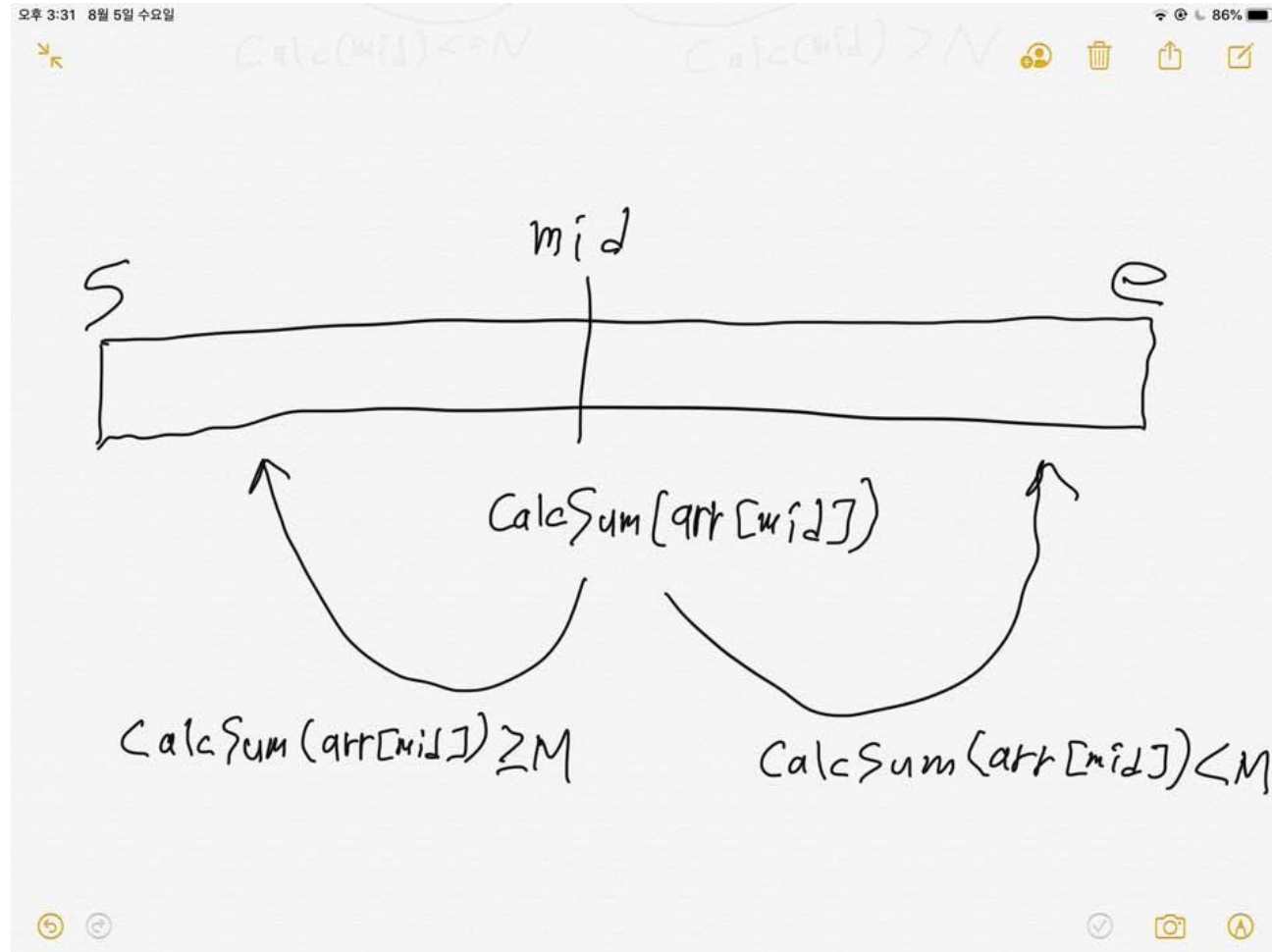
- 이분탐색내에 if(calcSum(arr[mid]) < M)는 calcSum 즉 해당 arr[mid]을예산의 최댓값으로 산정했을 때 전체 값(calcSum(~))이

M보다 작으면 arr[mid]보다 더 큰 한계값으로 이동하고

M보다 작거나 같으면 arr[mid]보다 작은 한계값으로 이동합니다.

- 이분탐색으로 찾은 적절한 한계값(arr[mid] 즉 maximum)에 대해서 maximum을 하나씩 줄어가면서 더 최적화 된 값을 찾거나 여기서 다시 세분화된 이분탐색을 사용할 수 도 있습니다.

boj-2512 예산



boj-2512 예산

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/2512.cpp
- <https://jaimemin.tistory.com/992>

boj-7453 합이 0인 네 정수
(난이도 : 상)

<https://www.acmicpc.net/problem/7453>

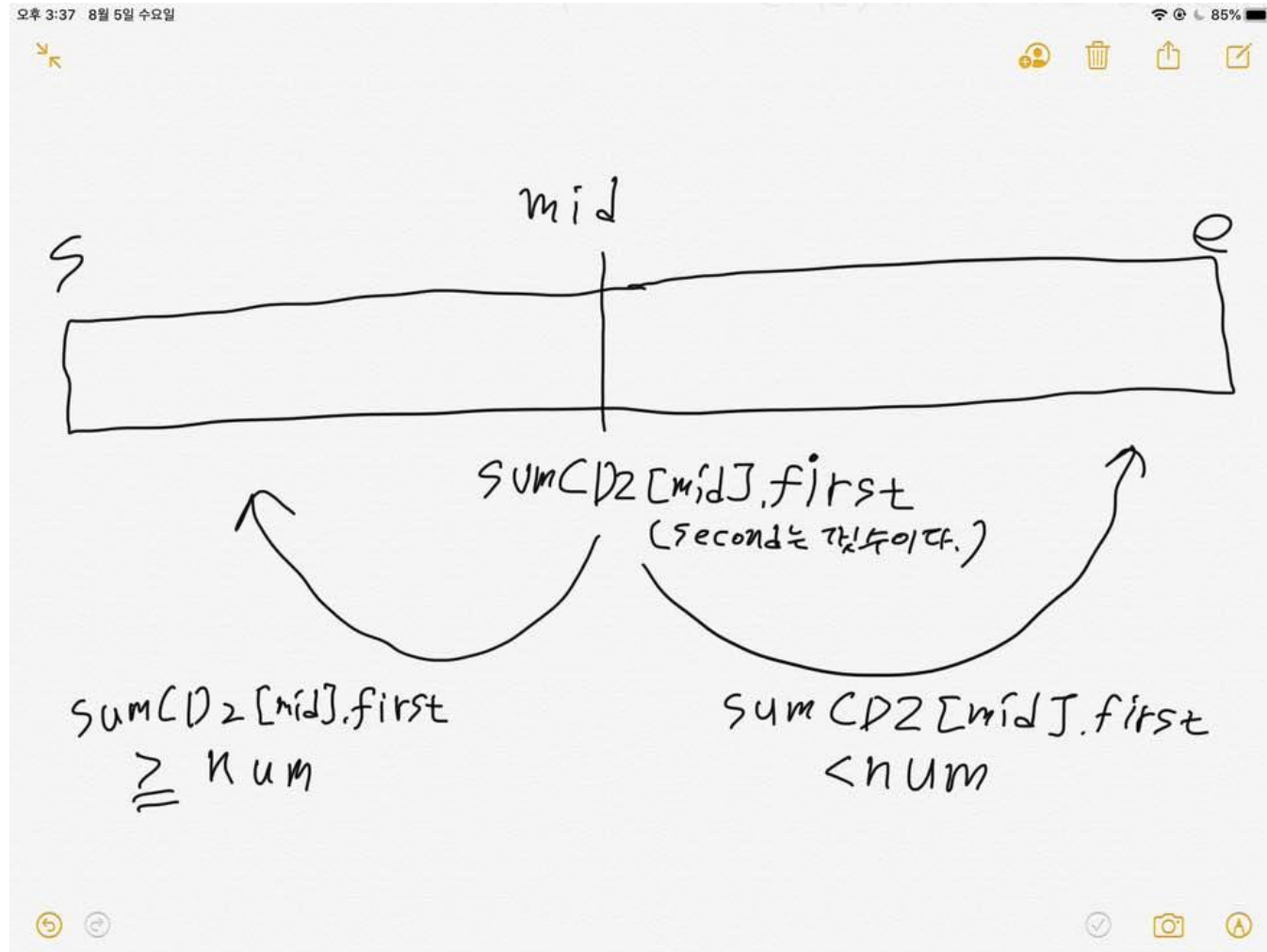
boj-7453 합이 0인 네 정수

- 배열 A,B,C,D의 합이 0인 index의 조합을 구하기 위해 이분탐색을 어떻게 사용해야 할지 고려한다.
- 각각의 4개 배열마다 이분탐색을 하면 뭔가 비효율적이다.
- 이분탐색의 장점은 아무리 많은 숫자들 사이에서도 $\log N$ 만에 찾아낸다는 장점을 활용해야한다.

boj-7453 합이 0인 네 정수

- 배열 A,B의 모든 가능한 경우의 합은 sumAB에 넣어준다.
- 배열 C,D의 모든 가능한 경우의 합은 sumCD에 넣어준다.
- 각각 최대 4000×4000 의 size를 갖는다.
- 이때 $-\text{sumAB} == \text{sumCD}$ 인 값을 sumCD내에서 이분탐색을 통해 찾는다.
- 시간복잡도는 $4000 \times 4000 \times \log(4000 \times 4000)$ 으로 상당히 줄어든다.

boj-7453 합이 0인 네 정수



boj-7453 합이 0인 네 정수

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/KiwanKim/7453.cpp
- [https://jaimemin.tistory.com/1108\(lower_bound,upper_bound\)](https://jaimemin.tistory.com/1108(lower_bound,upper_bound))
- 영훈이 질문 - 시간초과, <https://www.acmicpc.net/board/view/53837>

boj-2805 나무 자르기 (난이도 : 중)

<https://www.acmicpc.net/problem/2805>

boj-2805 나무 자르기

- H가 높을수록, 가져가는 나무의 양은 줄어듦
- H의 최소, 최대 = $[0, 1e+9]$
- 이분탐색으로 H의 범위를 줄여나가기
- 질문: H 범위의 중간값 mid을 구했을 때, 절단기의 높이가 mid면 나무 M미터 이상을 가져갈 수 있는가?
 - YES => low에 mid를 대입
 - NO => high에 mid를 대입

boj-2805 나무 자르기

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week5/boj-2805-try1.cpp

boj-2110 공유기 설치 (난이도 : 중)

<https://www.acmicpc.net/problem/2110>

boj-2110 공유기 설치

- 위치 x 입력 후 정렬부터 합니다.
- 공유기 min 거리 $\Delta x = 1$
 - 같은 좌표를 가진 집이 두 개 이상 존재하지 않으므로 0이 아님!!
- 공유기 MAX 거리 $\Delta x = (\text{맨 오른쪽 위치}) - (\text{맨 왼쪽 위치})$
- 마찬가지로 이분탐색으로 Δx 범위를 줄여나가기!
- 질문: 두 개의 공유기 사이의 거리가 mid 이상일 수 있는가?
 - YES => low에 mid 대입
 - NO => high에 mid 대입
- 맨 왼쪽에 공유기를 1개 놓는다!

boj-2110 공유기 설치

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week5/boj-2805-try2.cpp

boj-15732 도토리 숨기기 (난이도 : 상)

<https://www.acmicpc.net/problem/15732>

boj15732 - 도토리 숨기기

- 최소 상자 번호 = 1
- 최대 상자 번호 = N
- 역시나 이분탐색을 사용
- K개의 규칙을 모두 검사해서 카운팅
- 질문 : Count가 현재 도토리 개수 D 이상인가?
 - YES => high에 mid 대입
 - NO => low에 mid 대입

boj15732 - 도토리 숨기기

- https://github.com/alps-jbnu/ALPS_2020_Summer_Study/blob/master/Jubin/week5/boj-15732-try2.cpp

다음 6주차에는..

- 깊이 우선 탐색(DFS)을 배웁니다.
- 3주차 숙제를 오늘 제공합니다.
- 감사합니다.