SPC 2025

Sogang Programming Contest 2025

# 서강대학교 프로그래밍 대회

## 에디토리얼




주최

서강대학교 SOGANG UNIVERSITY

서강대학교 소프트웨어융합대학원

주관

Sogang ICPC Team

후원

과학기술정보통신부

IITP 정보통신기획평가원

Samsung Software Membership

STARTLINK

# Master

| 문제 | | 의도한 난이도 | 출제자 |
|---|---|---|---|
| **A** | 공강 사수 | **Easy** | 이재찬 dlwocks1209 |
| **B** | 효구장 | **Easy** | 김동건 dong_gas |
| **C** | dongshop | **Easy** | 김동건 dong_gas |
| **D** | 세미-연속 수열 | **Easy** | 김태윤 teyn |
| **E** | 근수의 카드게임 | **Medium** | 송근수 hulahula3247 |
| **F** | 원빈이의 인생 스케줄링 | **Medium** | 송근수 hulahula3247 |
| **G** | 최댓값 찾기 | **Medium** | 김규민 simsim07 |
| **H** | Collecting Sogang | **Hard** | 조원빈 wbcho0504 |
| **I** | 경숲길 재개발 2 | **Hard** | 이재찬 dlwocks1209 |



# Master A. 공강 사수

`implementation`
출제진 의도 – **Easy**

- ✓ 제출 76번, 정답 37명 (정답률 48.684%)
- ✓ 처음 푼 사람: **개굴개굴개구리**, 1분
- ✓ 출제자: 이재찬 `dlwocks1209`



- ✓ 지문 그대로 코드를 구현하면 됩니다.
- ✓ 구현 방법은 여러 가지가 있지만 한 예시로, 각 요일이 나왔는 지 확인하는 5개의 bool 변수를 만들고 모든 bool값을 확인하면 됩니다.
- ✓ set 자료구조에 넣고 길이가 5인지 확인하면 더 간단하게 구현이 가능합니다.



# MasterB. 효구장

`bruteforce`
출제진 의도 – **Easy**

- 제출 175번, 정답 22명 (정답률 12.571%)
- 처음 푼 사람: **nwc1718**, 5분
- 출제자: 김동건 `dong_gas`



- 한우팩 1개를 고르는 경우는 쉽게 계산할 수 있습니다.
- 한우팩 2개를 고르는 경우 역시 2중 for문으로 쉽게 계산할 수 있습니다.
- 한우팩 5개를 고르는 경우는 총합이기 때문에 쉽게 계산할 수 있습니다.
- 한우팩 3개를 고르는 경우도 3중 for문으로 계산할 수 있지만, 2개를 고르는 코드를 이용하여 전체 합에서 2개의 무게를 빼면 쉽게 계산할 수 있습니다.
- 한우팩 4개를 고르는 경우도 4중 for문으로 계산할 수 있지만, 전체 합에서 1개의 무게를 빼면 쉽게 계산할 수 있습니다.
- bit나 백트래킹을 이용하여 전체 한우팩을 고르는 모든 경우인 $2^5 - 1 = 31$가지를 계산하는 방법도 있습니다.



# MasterC.dongshop

`constructive`

출제진 의도 – **Easy**

- 제출 71번, 정답 27명 (정답률 38.028%)
- 처음 푼 사람: **자고싶다**, 14분
- 출제자: 김동건 `dong_gas`



- $N$개의 비숍을 모두 $\left\lfloor \frac{N}{2} \right\rfloor$ 행에 놓으면 모든 칸을 색칠할 수 있습니다.
- $\left\lfloor \frac{N}{2} \right\rfloor$ 행에 비숍 $N$개를 놓았다고 가정해봅시다.
- 격자판 위 임의의 칸 $(r, c)$에 대해 $d = \left\lfloor \frac{N}{2} \right\rfloor - r$이라 합시다.
- 이때, $(r + d = \left\lfloor \frac{N}{2} \right\rfloor, c + d)$와 $(r + d = \left\lfloor \frac{N}{2} \right\rfloor, c - d)$ 중 반드시 하나의 점은 $\left\lfloor \frac{N}{2} \right\rfloor$ 행 위의 점입니다.
- $1 \le c \le N$, $\left\lfloor \frac{N}{2} \right\rfloor - N \le d \le \left\lfloor \frac{N}{2} \right\rfloor - 1$ 이기 때문입니다.
- 따라서, $\left\lfloor \frac{N}{2} \right\rfloor$ 행에 비숍 $N$개를 놓았다면, 격자판 위의 임의의 칸 $(r, c)$는 항상 색칠되어 있습니다.



# Common A. 세미-연속 수열

`data_structure`, `tree_set`, `sliding_windows`
출제진 의도 – **Easy**

- ✓ Master 제출 93번, 정답 7명 (정답률 7.527%)
- ✓ Champion 제출 22번, 정답 6명 (정답률 27.273%)
- ✓ Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 27분
- ✓ Champion 처음 푼 사람: **나는꼬두람이**, 15분
- ✓ 출제자: 김태윤 `teyn`



- ✓ 목표를 다시 한번 생각해봅시다. 길이가 $K$인 연속 부분 수열이 $[p, p+1, p+2, \dots, p+k-1]$의 원소들이 중복 없이 이루어져 있는지 판별해야 합니다.
- ✓ 구간에 있는 원소가 하나하나 대응되는지 확인하는 것은 매우 비효율적임을 알 수 있습니다.
- ✓ 여기서 한번 고민을 해봅시다. 저희가 과연 모든 원소의 존재를 확인해야 할까요?
- ✓ 순열에서 연속된 자연수로 이루어져 있단 이야기는, 길이 $K$의 구간에서 $\max(v[i:i+k]) - \min(v[i:i+k])$의 값이 $K-1$임과 필요충분조건임을 알 수 있습니다.
- ✓ 목표를 재정의합시다. 모든 길이 $K$인 구간에서, 최댓값과 최솟값을 찾아 각 값의 차이가 $K-1$이 되는지 확인하는 문제로 변환되었습니다.

## Common A. 세미-연속 수열



- 그러나, 길이 $K$의 구간의 최댓값과 최솟값을 매번 확인하는 것은 $O((n-k)\cdot k)$의 시간복잡도를 가져 시간초과가 납니다
- 따라서, 구간에서의 최댓값/최솟값을 빠르게 찾을 수 있는 자료구조를 선정해야 합니다.
- 가장 쉬운 자료구조로는, $O(\log n)$만에 삽입, 삭제, 최댓값/최솟값 확인이 가능한 set을 이용하는 것입니다.
- set의 크기를 $K$로 유지하며, 스위핑하듯히 훑고 매 회마다 최대와 최소의 차이를 비교하여 답을 구할 수 있습니다.
- 이 외에도, 덱을 이용한 구간 최댓값 트릭, 세그먼트 트리 등의 자료구조 혹은 슬라이딩 윈도우를 이용한 에드혹 풀이 또한 가능합니다.



# Master E. 근수의 카드게임

`game_theory`, `ad_hoc`
출제진 의도 – **Medium**

- ✓ 제출 40번, 정답 15명 (정답률 37.500%)
- ✓ 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 40분
- ✓ 출제자: 송근수 `hulahula3247`



- ✓ $K < 2N$이라면 승형이는 항상 $1$만을 밴하는것이 최적입니다.
- ✓ 그러면 근수는 $2$ 또는 $3$만을 $N$번 선택해야 해서 최종 점수는 $2N$과 $3N$사이입니다. 이 값은 $K$보다 항상 높으므로 정답은 -1입니다.
- ✓ $K > 2N$이라면 승형이는 항상 $3$만을 밴하는것이 최적입니다.
- ✓ 그러면 근수는 $1$ 또는 $2$만을 $N$번 선택해야 해서 최종 점수는 $N$과 $2N$사이입니다. 따라서 근수는 항상 $2$를 고르는 것이 최적이고 정답은 $2N$입니다.

## Master E. 근수의 카드게임



- $K = 2N$이라면 승형이는 첫 턴에만 $2$를 밴하는 것이 최적입니다.
- 그러면 근수는 $1$ 혹은 $3$을 골라야 하는데, $3$을 선택한다면, 승형이는 다음턴부터 항상 $1$을 밴하기 시작해서 최종 점수가 $K$보다 높아집니다.
- 따라서, 근수는 $1$을 선택해야만 합니다.
- 그 이후에는 $K > 2N$인 상황과 동일하게 승형이는 항상 $3$을 밴하고 근수는 항상 $2$를 고릅니다.
- 따라서 이 경우에 정답은 $2N - 1$입니다.

# Common B. 원빈이의 인생 스케줄링

`prefix_sum`, `greedy`

출제진 의도 – **Medium**

- ✓ Master 제출 15번, 정답 4명 (정답률 26.667%)
- ✓ Champion 제출 8번, 정답 3명 (정답률 37.500%)
- ✓ Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 64분
- ✓ Champion 처음 푼 사람: **KHK**, 52분
- ✓ 출제자: 송근수 `hulahula3247`



- ✓ 우선 어떠한 작업 $i$에 대해서, $T_i < L_i$이면 원빈이는 해당 작업을 완료햐지 못하므로 정답은 $-1$입니다.
- ✓ 그 외의 경우에는, 모든 작업을 완료할 수 있는 선에서 greedy하게 운동 수치를 올리면 됩니다.
- ✓ 모든 시점마다 모든 작업과 비교를 하면 연산량 $O(200\,000^2)$로 너무 많습니다.
- ✓ 따라서 predix sum과 같은 자료구조로 전처리를 한번 해 두어서 각 시점마다 운동 수치를 올릴지 지식 수치를 올릴지 빠르게 결정하면 됩니다.

# Common C. 최댓값 찾기

`binary_search`, `math`

출제진 의도 – **Medium**

- Master 제출 17번, 정답 4명 (정답률 23.529%)
- Champion 제출 8번, 정답 3명 (정답률 37.500%)
- Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 77분
- Champion 처음 푼 사람: **KHK**, 75분
- 출제자: 김규민 `simsim07`

## Common C. 최댓값 찾기



- $X$를 가장 큰 수로, $f(t)$를 $\sum_{i=1}^{N} |t - x_i|$로 정의합시다.
- $X$이상인 모든 $t$에 대하여 $f(X)$는 $\sum_{i=1}^{N} t - x_i$이므로 $f(t) - f(X)$값은 $N \times (t - X)$값과 같습니다.
- 따라서, 임의의 $t$에 대하여 $f(10^5) - f(t)$의 값이 $N \times (10^5 - t)$와 같다면 $X$가 $t$이하이고, 그렇지 않다면 $X$가 $t$보다 큽니다.
- 이분탐색으로 $f(10^5) - f(t) = N \times (10^5 - t)$를 만족하는 가장 작은 $t$값을 구하여 정답을 찾을 수 있습니다.



# Common D. Collecting Sogang

`binary search`, `greedy`

출제진 의도 – **Hard**

- Master 제출 34번, 정답 0명 (정답률 0%)
- Champion 제출 4번, 정답 0명 (정답률 0%)
- 처음 푼 사람: ,0분
- 출제자: 조원빈 `wbcho0504`



- ✓ 이분탐색을 이용합니다. 그러면 기존의 문제를 $sogang$ 단어를 $T$개 만들 수 있는가라는 결정 문제로 바꿀 수 있습니다.
- ✓ 문자열의 앞에서부터 보며 $T$개의 $sogang$을 만듭니다.
- ✓ $g$를 제외한 문자는 사용할 수 있는 경우 사용합니다.
- ✓ 하지만 $g$의 경우, $so$ 뒤의 $g$로 사용할 것인지 아니면 $sogan$ 뒤의 $g$로 사용할 것인지가 모호합니다.



- ✓ 여기서 $sog$ $T$개를 먼저 완성하는 그리디가 성립합니다.
- ✓ 구체적으로 $g$가 등장한 경우,
  1. 지금까지 만든 $sog$의 개수가 $T$개 이하이고 $so$가 존재하면, 그 뒤에 $g$를 붙여 $sog$를 만듭니다.
  2. 위의 경우를 만족하지 않으면서 $sogan$이 존재하면, 그 뒤에 $g$를 붙여 $sogang$을 만듭니다.
- ✓ 위 방법을 이용해 $O(N \log N)$ 에 문제를 해결할 수 있으며, 이를 응용한 선형 풀이도 존재합니다.



# MasterI. 경숲길 재개발 2

`greedy, stack`

출제진 의도 – **Hard**

- ✓ 제출 1번, 정답 0명 (정답률 0%)
- ✓ 처음 푼 사람: , n분
- ✓ 출제자: 이재찬 `dlwocks1209`



- ✓ 이 문제는 그리디 풀이가 성립합니다. 왼쪽부터 차례대로 채우면서, 항상 지정할 수 있는 가장 작은 높이로 설정하는 것이 최적해임이 보장됩니다.
- ✓ 그리디 해법을 엄밀히 설명하자면 다음과 같습니다.
- ✓ 현재 확인중인 칸 $i$에서 $H_i$가 $0$일 때, 다음 두 가지 케이스가 존재합니다.
- ✓ $i$가 마지막 칸이거나 $H_{i+1}$가 $0$이라면 왼쪽 건물들의 높이만을 고려하여 지정 가능한 가장 작은 높이로 설정합니다.
- ✓ $H_{i+1}$에 이미 값이 채워져 있다면 왼쪽 건물들의 높이와 $i+1$부터 다음으로 나오는 첫 $0$까지의 건물 높이를 고려하여 지정 가능한 가장 작은 높이로 설정합니다.



- 모든 $H_i = 0$인 $i$에 대해 왼쪽 건물들의 높이를 완전 탐색으로 확인한다면 전체 시간복잡도는 $O(N^2)$이 됩니다.
- 따라서 왼쪽 건물들의 높이 중 필수적인 정보만을 저장해야 합니다.
- 임의의 $x, y$에 대해 $x < y$면서 $H_x < H_y$라면 $x$는 $y$오른쪽에 있는 칸들에게 아무런 영향력을 줄 수 없습니다.
- 이 사실을 이용하면 $H_i$의 값을 정할 때, 왼쪽 건물들 중 감소하는 부분 수열로 이루어진 일부만 기억하면 됩니다.
- 이는 모노톤 스택을 이용하여 구현 가능합니다. 새로운 값을 넣을 때, 스택의 top이 이 값보다 작다면 top이 더 커질 때까지 스택을 pop한 후 새로운 값을 넣으면 됩니다.



- ✓ 가장 높은 건물의 높이가 $M$이라면 스택의 길이는 최대 $M$이고 전체 시간복잡도는 $O(NM)$이 됩니다.
- ✓ 초기에 주어지는 건물의 최대 높이가 $100$이기 때문에, $M$은 최대 $100 + \lceil \log(N) \rceil$이며, 시간 내에 해결이 가능합니다.



- ✓ 그리디 풀이가 성립한다는 것은 교환 논증을 이용해 증명할 수 있습니다.
- ✓ 그리디 풀이 $G$가 아닌 최적해 $OPT$가 존재한다고 가정해봅시다.
- ✓ 그렇다면 $H_i = 0$인 어떤 $i$에 대해, $i$ 이전의 값들은 $G$와 $OPT$가 전부 동일하고, $G$에서 설정한 $H_i$보다 $OPT$에서 설정한 $H_i$가 더 큰 $i$가 존재하게 됩니다.
- ✓ $i$가 마지막으로 등장하는 $0$이라면 $G$가 $OPT$보다 무조건 유리하기 때문에 $i < j, H_j = 0$인 $j$가 존재하는 경우만 확인하면 됩니다.



- ✓ $G$의 $H_i$를 $x$, $OPT$의 $H_i$를 $y$로 둘 수 있습니다. $(x < y)$
- ✓ 우선 $i$와 $j$의 차가 1인 경우를 봅시다.
- ✓ $OPT$에서 $H_j$가 $z(z < y)$ 라면 $G$에서 $H_j$를 $y$로 두기만 해도 $OPT$보다 유리해집니다.
- ✓ 이 때 $i$ 위치에 $y$가 들어올 수 있었기에, $G$의 $H_j$에도 항상 $y$로 설정이 가능합니다.
- ✓ $OPT$에서 $H_j$가 $w(w > y)$ 라면 $G$에서도 $H_j$를 $w$로 설정할 수 있고, $OPT$는 $G$와 같아집니다.



- 다음으로 $i$와 $j$ 사이에 $0$이 아닌 값이 있는 경우를 봅시다.
- $i$와 $j$사이에 있는 모든 건물 높이가 $y$보다 작다면 $G$에서 $H_j$를 $y$로 설정 가능합니다. 이 경우 이전의 경우와 비슷하게 $G$가 유리해집니다.
- $i$와 $j$사이에 있는 건물 중 $y$보다 높은 것이 있다면 이 순간 $G$와 $OPT$는 같아집니다.
- 결국 모든 경우에 대해 $OPT$는 $G$와 같거나 불리하기 때문에, 즉 $OPT$가 최적해라는 것은 모순이기 때문에 증명이 성립합니다.

# Champion

| 문제 | | 의도한 난이도 | 출제자 |
|---|---|---|---|
| **A** | 크기가 4인 박스 | **Easy** | 송근수 hulahula3247 |
| **B** | 세미-연속 수열 | **Easy** | 김태윤 teyn |
| **C** | 트리펴기 | **Medium** | 김동건 dong_gas |
| **D** | 경숲길 재개발 1 | **Medium** | 이재찬 dlwocks1209 |
| **E** | 원빈이의 인생 스케줄링 | **Medium** | 송근수 hulahula3247 |
| **F** | 최댓값 찾기 | **Medium** | 김규민 simsim07 |
| **G** | Collecting Sogang | **Hard** | 조원빈 wbcho0504 |
| **H** | 트리 초기화 | **Hard** | 이재찬 dlwocks1209 |
| **I** | 배수로 뚫기 | **Hard** | 조원빈 wbcho0504 |



# Champion A. 크기가 4인 박스

`greedy`
출제진 의도 – **Easy**

- ✓ 제출 40번, 정답 14명 (정답률 35.000%)
- ✓ 처음 푼 사람: **나는꼬두람이**, 4분
- ✓ 출제자: 송근수 `hulahula3247`

## **Champion A**. 크기가 4인 박스



- 크기가 $4$인 물건은 따로 따로 담아야만 합니다.
- 크기가 $3$인 물건은 크기가 $1$인 물건과 같이 담을 수 있습니다. 크기가 $1$인 물건이 존재한다면 같이 담으면 되고, 존재하지 않으면 따로 담으면 됩니다.
- 크기가 $2$인 물건은 크기가 $2$인 물건과 같이 담을 수 있습니다. 존재하지 않는다면 크기가 $1$인 물건들과 같이 담으면 됩니다.
- 이제 크기가 $1$인 물건들만 남았습니다. 크기가 $4$인 박스에 최대한 합쳐서 담으면 됩니다.



# Common A. 세미-연속 수열

`data_structure`, `tree_set`, `sliding_windows`
출제진 의도 – **Easy**

- ✓ Master 제출 93번, 정답 7명 (정답률 7.527%)
- ✓ Champion 제출 22번, 정답 6명 (정답률 27.273%)
- ✓ Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 27분
- ✓ Champion 처음 푼 사람: **나는꼬두람이**, 15분
- ✓ 출제자: 김태윤 `teyn`

## **Common A**. 세미-연속 수열



- 목표를 다시 한번 생각해봅시다. 길이가 $K$인 연속 부분 수열이 $[p, p+1, p+2, \dots, p+k-1]$의 원소들이 중복 없이 이루어져 있는지 판별해야 합니다.
- 구간에 있는 원소가 하나하나 대응되는지 확인하는 것은 매우 비효율적임을 알 수 있습니다.
- 여기서 한번 고민을 해봅시다. 저희가 과연 모든 원소의 존재를 확인해야 할까요?
- 순열에서 연속된 자연수로 이루어져 있단 이야기는, 길이 $K$의 구간에서 $\max(v[i:i+k]) - \min(v[i:i+k])$의 값이 $K-1$임과 필요충분조건임을 알 수 있습니다.
- 목표를 재정의합시다. 모든 길이 $K$인 구간에서, 최댓값과 최솟값을 찾아 각 값의 차이가 $K-1$이 되는지 확인하는 문제로 변환되었습니다.

## Common A. 세미-연속 수열



- 그러나, 길이 $K$의 구간의 최댓값과 최솟값을 매번 확인하는 것은 $O((n-k)\cdot k)$의 시간복잡도를 가져 시간초과가 납니다
- 따라서, 구간에서의 최댓값/최솟값을 빠르게 찾을 수 있는 자료구조를 선정해야 합니다.
- 가장 쉬운 자료구조로는, $O(\log n)$만에 삽입, 삭제, 최댓값/최솟값 확인이 가능한 set을 이용하는 것입니다.
- set의 크기를 $K$로 유지하며, 스위핑하듯히 훑고 매 회마다 최대와 최소의 차이를 비교하여 답을 구할 수 있습니다.
- 이 외에도, 덱을 이용한 구간 최댓값 트릭, 세그먼트 트리 등의 자료구조 혹은 슬라이딩 윈도우를 이용한 에드혹 풀이 또한 가능합니다.



# Champion C. 트리 펴기

`ad_hoc`, `trees`

출제진 의도 – **Medium**

- 제출 23번, 정답 9명 (정답률 43.478%)
- 처음 푼 사람: **KHK**, 20분
- 출제자: 김동건 `dong_gas`

## Champion C. 트리 펴기



- 해설에 쓰인 변수 $s, t, p, S, T$의 의미는 문제 지문에서 확인할 수 있습니다.
- 차수가 $3$이상인 노드들의 집합을 $X$라 하고, $X$에 속한 노드들의 $(-2)$의 합을 $sum$이라 합시다.
- 반대로 차수가 $2$이하인 노드들의 집합을 $Y$라 합시다.
- 우리의 목표는 트리의 모든 노드의 차수가 $2$이하가 되게 만드는 것입니다. 이는 $X$를 공집합으로 만드는 것과 동치이고, 이는 $sum$을 $0$으로 만드는 것과도 동치입니다.

## Champion C. 트리 펴기



- 매 작업마다 아래의 규칙을 지키며 작업을 진행해 봅시다.
- 1. 우리는 매 작업마다 차수가 $1$이면서, 인접한 노드의 차수가 $3$ 이상인 노드를 $s$로 선택하고, $s$와 인접한 유일한 노드를 $t$로 선택하여 트리를 $S$와 $T$로 나눕니다.
- 2. 트리 $T$의 노드 중 차수가 $1$인 노드 중 하나를 $p$로 선택하고, $s$와 $p$를 잇습니다.

## Champion C. 트리 펴기



- ✓ 작업을 통해 $p$는 차수가 $1$에서 $2$로 변하기 때문에 $sum$에 영향을 주지 않습니다.
- ✓ 작업을 통해 $t$는 $3$ 이상이던 차수가 $1$만큼 줄어들기 때문에, $sum$을 $1$만큼 감소시키는 효과가 있습니다.
- ✓ $s$를 포함한 나머지 노드들은 차수에 변화가 없음이 자명합니다.

## Champion C. 트리 펴기



- ✓ 한 번의 작업에서 차수가 줄어들 수 있는 노드는 $t$가 유일합니다. 따라서, 어떠한 작업으로도 $sum$을 한 번에 $2$ 이상 감소시키는 방법은 존재하지 않습니다.
- ✓ 반면, 우리가 정의한 규칙대로 작업을 진행하면 한 번의 작업에서 무조건 $sum$을 $1$만큼 감소시킬 수 있으므로 이를 반복하는 것이 최적해입니다.
- ✓ 작업은 차수가 $3$ 이상인 노드가 사라질 때까지 해야 하므로 작업 횟수는 $sum$과 동일하고, 이는 $\sum_{i=1}^{N} max(0, d_i - 2)$입니다. ($d_i$는 노드 $i$의 차수)



# Champion D. 경숲길 재개발 1

`Divide_and_Conquer`, `dp`

출제진 의도 – **Medium**

- ✓ 제출 31번, 정답 6명 (정답률 19.355%)
- ✓ 처음 푼 사람: **KHK**, 40분
- ✓ 출제자: 이재찬 `dlwocks1209`

## **Champion D.** 경숲길 재개발 1



- ✓ 이 문제의 최적해는 왼쪽부터 채울 수 있는 가장 작은 값을 채워넣는 것입니다.
- ✓ 최적해는 1, 2, 1, 3, 1, 2, 1, 4 ... 와 같은 형태가 됩니다.
- ✓ $N$이 최대 $10^12$이기 때문에 선형 탐색으로는 해결할 수 없습니다. $log(N)$의 시간에 해결하기 위해 분할정복과 DP를 이용해야 합니다.

## Champion D. 경숲길 재개발 1



- 건물의 수가 $N$일 때 모든 건물 높이의 합을 $f(N)$이라 하면 다음과 같이 점화식을 세울 수 있습니다.
- $k$를 건물의 수가 $N$일 때 가장 높은 건물의 위치라고 했을 때($k$는 유일)
  $f(N) = f(k-1) + H_k + f(N-k)$이 됩니다. 이는 $k$를 기준으로 왼쪽과 오른쪽을 나누는 것을 의미합니다.
- 이 과정을 재귀적으로 반복하면서 메모이제이션을 하면 $log(N)$의 시간복잡도에 해결이 가능합니다.



- 여기서 $k$는 다음 과정을 통해 구할 수 있습니다.
- $g(X)$를 높이가 $X$ 이하인 건물들로 만들 수 있는 최대 건물 수라고 했을 때 $g(X+1)$은 높이가 $X+1$인 건물 양 옆에 $g(X)$둔 형태입니다.
- 식으로 나타내면 $g(N+1) = 2g(N) + 1$이고, 일반항은 $g(X) = 2^X - 1$입니다.
- 이것의 역함수인 로그를 이용하면 건물 개수 $N$을 이용하여 가장 높은 건물의 높이를 구할 수 있고, k를 구할 수 있습니다.

# Common B. 원빈이의 인생 스케줄링

`prefix_sum`, `greedy`

출제진 의도 – **Medium**

- Master 제출 15번, 정답 4명 (정답률 26.667%)
- Champion 제출 8번, 정답 3명 (정답률 37.500%)
- Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 64분
- Champion 처음 푼 사람: **KHK**, 52분
- 출제자: 송근수 `hulahula3247`



- ✓ 우선 어떠한 작업 $i$에 대해서, $T_i < L_i$이면 원빈이는 해당 작업을 완료햐지 못하므로 정답은 $-1$입니다.
- ✓ 그 외의 경우에는, 모든 작업을 완료할 수 있는 선에서 greedy하게 운동 수치를 올리면 됩니다.
- ✓ 모든 시점마다 모든 작업과 비교를 하면 연산량 $O(200\,000^2)$로 너무 많습니다.
- ✓ 따라서 predix sum과 같은 자료구조로 전처리를 한번 해 두어서 각 시점마다 운동 수치를 올릴지 지식 수치를 올릴지 빠르게 결정하면 됩니다.



# Common C. 최댓값 찾기

`binary_search`, `math`

출제진 의도 – **Medium**

- Master 제출 17번, 정답 4명 (정답률 23.529%)
- Champion 제출 8번, 정답 3명 (정답률 37.500%)
- Master 처음 푼 사람: **내가노주환을이길수있을리없잖아무리무리**, 77분
- Champion 처음 푼 사람: **KHK**, 75분
- 출제자: 김규민 `simsim07`

## Common C. 최댓값 찾기



- ✓ $X$를 가장 큰 수로, $f(t)$를 $\sum_{i=1}^{N} |t - x_i|$로 정의합시다.
- ✓ $X$이상인 모든 $t$에 대하여 $f(X)$는 $\sum_{i=1}^{N} t - x_i$이므로 $f(t) - f(X)$값은 $N \times (t - X)$값과 같습니다.
- ✓ 따라서, 임의의 $t$에 대하여 $f(10^5) - f(t)$의 값이 $N \times (10^5 - t)$와 같다면 $X$가 $t$이하이고, 그렇지 않다면 $X$가 $t$보다 큽니다.
- ✓ 이분탐색으로 $f(10^5) - f(t) = N \times (10^5 - t)$를 만족하는 가장 작은 $t$값을 구하여 정답을 찾을 수 있습니다.



# Common D. Collecting Sogang

`binary search`, `greedy`

출제진 의도 – **Hard**

- Master 제출 34번, 정답 0명 (정답률 0%)
- Champion 제출 4번, 정답 0명 (정답률 0%)
- 처음 푼 사람: ,0분
- 출제자: 조원빈 `wbcho0504`



- ✓ 이분탐색을 이용합니다. 그러면 기존의 문제를 $sogang$ 단어를 $T$개 만들 수 있는가라는 결정 문제로 바꿀 수 있습니다.
- ✓ 문자열의 앞에서부터 보며 $T$개의 $sogang$을 만듭니다.
- ✓ $g$를 제외한 문자는 사용할 수 있는 경우 사용합니다.
- ✓ 하지만 $g$의 경우, $so$ 뒤의 $g$로 사용할 것인지 아니면 $sogan$ 뒤의 $g$로 사용할 것인지가 모호합니다.



- ✓ 여기서 $sog$ $T$개를 먼저 완성하는 그리디가 성립합니다.
- ✓ 구체적으로 $g$가 등장한 경우,
  1. 지금까지 만든 $sog$의 개수가 $T$개 이하이고 $so$가 존재하면, 그 뒤에 $g$를 붙여 $sog$를 만듭니다.
  2. 위의 경우를 만족하지 않으면서 $sogan$이 존재하면, 그 뒤에 $g$를 붙여 $sogang$을 만듭니다.
- ✓ 위 방법을 이용해 $O(N \log N)$ 에 문제를 해결할 수 있으며, 이를 응용한 선형 풀이도 존재합니다.



# Champion H. 트리 초기화

`tree_dp`, `graph_traversal`
출제진 의도 – **Hard**

- ✓ 제출 6번, 정답 1명 (정답률 16.667%)
- ✓ 처음 푼 사람: **나는꼬두람이**, 154분
- ✓ 출제자: 이재찬 `dlwocks1209`

## Champion H. 트리 초기화



- 가중치의 흐름은 닫혀있기 때문에 2번 연산을 수행해도 가중치 총량은 변하지 않습니다.
- 따라서 해가 존재하려면 모든 가중치의 합이 0이 되어야 합니다.
- 해가 존재하는 트리에서 특정 간선을 기준으로 왼쪽 서브트리와 오른쪽 서브트리를 나눈다면 각 서브트리의 가중치 합은 절댓값이 같고 부호가 반대인 값이 됩니다.
- 그리고 이 부호의 방향이 해당 간선에서 가중치가 이동해야 하는 방향이며, 절댓값이 이동해야 하는 가중치량입니다.
- 간선에서 해당 방향으로 가중치가 이동하려면 가중치를 보내는쪽 서브트리에서 루트가 최소 한 번 설정되어야 합니다.
- 트리의 모든 간선에 대해 이것이 수행되는 것이 트리 초기화의 필요충분조건입니다.



- 이것을 만족하는 최소 루트 설정 횟수는 다음 과정을 통해 구할 수 있습니다.
- 우선 트리의 모든 간선에 대해 가중치 흐름 방향을 결정합니다.
- 이는 tree dp를 이용하면 $O(N)$에 수행할 수 있습니다.
- 임의의 정점을 시작으로 후위 순회를 하면서 가중치 방향에 따라 필요한 정점만 루트로 지정하면 지정해야 할 루트 개수를 구할 수 있습니다.



# I. 연우의 배수로 뚫기

`stack`, `cartesian tree`, `segment tree`

출제진 의도 – **Hard**

- ✓ 제출 0번, 정답 0명 (정답률 0%)
- ✓ 처음 푼 사람: , n분
- ✓ 출제자: 조원빈 `wbcho0504`



- 먼저 초기 상태의 물의 총량은 다음의 방법으로 구할 수 있습니다.
- $i$번 칸의 입장에서 볼 때, 왼쪽 칸 중 높이의 최댓값을 $lmx$, 오른쪽 칸 중 높이의 최댓값을 $rmx$라고 하면, $i$번 칸에 있는 물의 양은 $max(min(lmx, rmx) - h_i, 0)$ 입니다.
- 물의 총량은 각 칸에 있는 물의 양의 합을 계산해 구할 수 있습니다.



- 물이 고이지 않은 칸들을 기준으로 보면, 높이가 단조 증가하다가 단조 감소하는 산 모양의 형태를 관찰할 수 있습니다.
- 배수로를 설치하면, 칸의 높이는 $0$이 되고 그 칸을 기준으로 2개의 컴포넌트로 분리되며, 나눠진 두 개의 컴포넌트 또한 산 모양의 형태를 띄게 됩니다.
- 이러한 특성을 이용해 배수로를 설치한 후 물의 총량의 변화를 빠르게 계산할 수 있습니다.



- 구체적으로 $i$번 칸에 배수로를 설치하는 경우, $i$번 칸의 높이는 $0$이 되고 왼쪽과 오른쪽에 고인 물의 변화를 갱신해주어야 합니다.
- $i-1$번 칸에서 시작해 $left$ 값을 참조해가며 변화를 업데이트합니다.
- $i+1$번 칸에서 시작해 $right$ 값을 참조해가며 변화를 업데이트합니다.
- 각 칸 $j$를 기준으로 보면, $left[j]$와 $right[j]$에 대해 총 두 번의 업데이트만 발생합니다. 따라서 업데이트는 최대 $2N$번 발생합니다.



- ✓ $Q \leq N$ 이므로 위 풀이를 통해 $O(N)$에 문제를 해결할 수 있습니다.
- ✓ cartesian tree나 sparse table 등을 이용해서도 문제를 해결할 수 있습니다.
- ✓ 구현 시 높이가 같은 경우의 처리에 유의합시다.