

# 2022 숭고한 연합 알고리즘 대회

## 공식 해설

## Sponsors



## Sponsors



## 검수

✓ 김세린	dennisstar	KAIST 수리과학과
✓ 모현	ahgus89	가톨릭대학교 의예과
✓ 박선재	cs71107	서울대학교 컴퓨터공학부
✓ 박원	chunghan	UNIST 컴퓨터공학과
✓ 송은하	runnie0427	KAIST 수리과학과
✓ 심유근	cozyyg	서울대학교 컴퓨터공학부
✓ 윤시우	cgiosy	

번호	문제	의도한 난이도	출제자
2A	자동 완성	Easy	SCCC 이로건
2B	장작 넣기	Easy	Alkor 강상수
2C	주식	Easy	SCCC 이영재
2D	SKH 문자열	Medium	SCCC 이영재
2E	최대한의 휴식	Medium	Alkor 김재우
2F	노트 조각	Medium	SCCC 김도현
2G	통행량 조사	Medium	SCCC 나정휘
2H	원형 게임	Hard	SCCC 오주원

번호	문제	의도한 난이도	출제자
1A	단어 마방진	Easy	ALOHA 방인규
1B	이차 함수	Medium	ALPS 노세윤
1C	캐슬 디펜스	Medium	ALPS 노세윤
1D	내적	Medium	ALPS 노세윤
1E	다트	Medium	ALOHA 이현빈
1F	르블랑의 트리 순회	Hard	SCCC 이성서
1G	숲 게임	Hard	SCCC 김도현
1H	DCMSF	Hard	SCCC 나정휘

## 2A. 자동 완성

implementation

출제진 의도 – **easy**

- ✓ 처음 푼 사람: **구재원**, 0분
- ✓ 출제자: SCCC 이로건

## 2A. 자동 완성

- ✓ 입출력과 조건문을 이용해 간단하게 해결할 수 있습니다.



## 2B. 장작 넣기

recursion, bruteforcing, simulation

출제진 의도 – **easy**

- ✓ 처음 푼 사람: **강근호**, 19분
- ✓ 출제자: Alkor 강상수

## 2B. 장작 넣기

- ✓ 조건을 만족하는 경우의 수를 구하는 구하는 문제이므로 완전 탐색을 시도해봅시다.
- ✓ 매 시각 모닥불의 화력이 감소하는 것은  $\mathcal{O}(N)$  시간에 처리할 수 있습니다.
- ✓ 화력이 감소한 뒤, 현재 위치, 왼쪽, 오른쪽 중 하나를 선택해 수행할 수 있습니다.

## 2B. 장작 넣기

- ✓ 매 시각마다 3개의 행동을 선택할 수 있으므로  $3^T$  가지 경우를 확인하면 됩니다.
- ✓ 완전 탐색의 시간 복잡도는  $\mathcal{O}(3^T \times N)$  입니다.
- ✓  $T \leq 11, N \leq 6$  이므로 완전 탐색을 이용해 문제를 해결할 수 있습니다.

## 2C. 주식

bruteforcing, dp  
출제진 의도 – **easy**

- ✓ 처음 푼 사람: **강근호**, 41분
- ✓ 출제자: SCCC 이영재

## 2C. 주식

- ✓ DP 또는 Brute force를 이용해 해결할 수 있습니다.

## 2C. 주식

### ✓ Brute force를 이용한 해결 방법

- 매일 아래 네 가지 행동 중 하나를 선택해야 합니다.
  1. 아무것도 안 함
  2. 대출을 최대한 많이 받아서 주식 매수
  3. 갖고 있는 주식 전부 매도
  4. 갖고 있는 주식을 전부 매도한 뒤 대출 받아서 매수
- 총  $\mathcal{O}(4^N)$  가지의 경우를 확인하면 되고, 매번  $\mathcal{O}(N)$  시간에 최종 금액을 구할 수 있습니다.
- 전체 시간 복잡도는  $\mathcal{O}(N \times 4^N)$  입니다.
- 계산할 때 몇 가지 유의해야 할 점이 있는데, DP 풀이에서 함께 살펴보겠습니다.

## 2C. 주식

### ✓ DP를 이용한 해결 방법

- $D_i$ 를 대출금을 다 갚은 상태로  $i$ 일에 가질 수 있는 최대 금액이라고 정의합니다.
- $D_i = \max_{0 \leq j < i} \left\{ \left\lfloor \frac{D_j(K+1)}{P_j} \right\rfloor \cdot P_i + (D_j(K+1) \bmod P_j) - P_j K \right\}$ 
  - ▶  $D_j(K+1)/P_j$ :  $j$ 일에 대출을 받아서 살 수 있는 주식의 최대 개수
  - ▶  $D_j(K+1) \bmod P_j$ :  $j$ 일에 대출을 받아서 주식을 최대한 많이 사고 남은 금액
  - ▶  $P_j K$ :  $j$ 일에 대출을 받은 금액
- 대출금을 갚지 않아도 되므로  $P_j K$ 를 빼지 않고 계산한 결과의 최댓값이 정답이 됩니다.
- 시간 복잡도는  $\mathcal{O}(N^2)$ 입니다.

## 2D. SKH 문자열

greedy

출제진 의도 – **medium**

- ✓ 처음 푼 사람: **김동우**, 159분
- ✓ 출제자: SCCC 이영재



## 2D. SKH 문자열

- ✓ 더 적은 문자를 사용해 SKH를 만드는 경우부터 확인하면 됩니다.
  1. SKH:  $S, K, H$ 를 추가로 사용하지 않는 되는 경우
  2. SK,SH,KH: 문자 한 개( $S, K, H$ )를 추가해서 SKH를 만드는 경우
  3. S, K, H: 문자 두 개(KH, SH, SK)를 추가해서 SKH를 만드는 경우
  4. 문자 세 개SKH를 추가해서 SKH를 만드는 경우

## 2D. SKH 문자열

- ✓ (1)은 단순히 SKH이 몇 번 등장하는지 확인하면 됩니다.
- ✓ (1)에서 사용한 문자들의 인덱스를 저장합니다.
- ✓ (2)는 (1)에서 사용하지 않은 인덱스로 SK, SH, KH를 몇 번 만들 수 있는지 확인하면 됩니다.
- ✓  $\mathcal{O}(N)$ 에 할 수 있습니다.

## 2D. SKH 문자열

- ✓ (3), (4)는 함께 처리합니다.
- ✓ S에 KH를 추가해서 SKH를 만드는 횟수  $t$ 를 고정해봅시다.
- ✓ 이때 사용되지 않은 K와 H는 이제 독립적으로 사용할 수 있습니다.
- ✓ K에 SH를 추가해서 SKH를 최대한 많이 만든 다음, H에 SK를 추가해서 SKH를 만듭니다.
- ✓ 이렇게 (3)을 처리할 수 있고, (4)는 위에서 다 사용하지 못한 SKH를 묶어주면 됩니다.
- ✓ 고정된  $t$ 에 대해, 위 과정을  $\mathcal{O}(1)$ 에 처리할 수 있습니다.
- ✓ KH의 개수 고정하는 경우의 수, SK를 고정하는 경우의 수, KH를 고정하는 경우의 수는 총  $\mathcal{O}(p + q + r)$ 가지입니다.
- ✓ 전체 문제를  $\mathcal{O}(N + p + q + r)$ 에 해결할 수 있습니다.

## 2E. 최대한의 휴식

parametric\_search, dp

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **구재원**, 144분
- ✓ 출제자: Alkor 김재우

## 2E. 최대한의 휴식

- ✓ 먼저, 예외처리를 해주어야 하는 경우에 대해 생각해봅시다.
  - -1을 출력해야 하는 경우는 주어진  $W_i$ 들의 총합이  $M$ 보다 작은 경우입니다.
  - "Free!"를 출력해야 하는 경우는 주어진  $W_i$ 들 중 최댓값이  $M$ 보다 크거나 같은 경우입니다.
  - 이 경우들에 대해서는 사전에 입력받으면서 예외 처리를 해주면 됩니다.

## 2E. 최대한의 휴식

- ✓ 이후, 문제 조건을 분석해보면 다음과 같은 이유에서 parametric search가 가능함을 알 수 있습니다.
  - $M$  이상의 일을 하면서 연휴들 중 최소값이  $k$  이상이 되도록 할 수 있다면,  $M$  이상의 일을 하면서 연휴들 중 최소값이  $k - 1$  이상이 되도록 할 수 있습니다.
  - $M$  이상의 일을 하면서 연휴들 중 최소값이  $k$  이상이 되도록 할 수 없다면,  $M$  이상의 일을 하면서 연휴들 중 최소값이  $k + 1$  이상이 되도록 할 수 없습니다.
- ✓ 따라서 binary search를 통해 정답의 범위를 줄여나가면서 정답을 구할 수 있습니다.
- ✓ 위 과정을  $\mathcal{O}(\log N)$  번 반복해야 합니다.

## 2E. 최대한의 휴식

- ✓ 어떤  $k$ 에 대해 연휴의 최소 길이가  $k$  이상이면서 일을  $M$  이상 할 수 있는지 판별해야 합니다.
- ✓ 이는 1차원 DP를 통해 해결할 수 있는데, 이를 위해 다음과 같이 DP식을 정의할 수 있습니다.
  - $DP[i]$  :  $i$  번째 날에 일을 하면서  $i$  번째 날 이전의 연휴들 중 최솟값이  $k$  이상이 되도록 할 때 할 수 있는 일의 최대 양
  - $DP[i] = \max_{0 \leq j < i-k} (DP[j]) + W_i$
- ✓  $DP$ 의 최댓값과  $M$ 의 값을 비교함으로써  $k$ 에 대해 조건을 만족하는지 판별할 수 있습니다.
- ✓ 주어진  $k$ 에 대해, 조건을 만족하는지 판별하는데 필요한 시간은  $\mathcal{O}(N)$ 입니다.
- ✓ 따라서  $\mathcal{O}(N \log N)$ 에 문제를 해결할 수 있습니다.

## 2F. 노트 조각

dijkstra, dp

출제진 의도 - **Medium**

- ✓ 처음 푼 사람: **구재원**, 70분
- ✓ 출제자: SCCC 김도현



## 2F. 노트 조각

- ✓ 1번 정점에서 출발해 모든 노트 조각을 방문하고  $N$  번 정점까지 가는 경로를 찾는 문제입니다.
- ✓ 이때 경로의 길이는 1번 정점에서  $N$  번 정점까지 가는 최단 거리와 동일해야 합니다.

## 2F. 노트 조각

- ✓ 노트 조각을 모두 지나는 경로 중 최소를 찾는 것은 TSP이기 때문에 어렵습니다.
- ✓ 최단 경로를 먼저 구해놓고 노트 조각을 모두 지나도록 약간씩 수정할 수 있을까요?
- ✓ 모든 최단 경로의 합집합을 생각해봅시다.
  - 가중치가 모두 양수이므로 사이클이 없습니다.
  - 이 그래프의 임의의 경로는 전부 기존 그래프의 최단 경로입니다.
- ✓ 이를 최단 경로 DAG라고 합니다. DAG의 최단 경로와는 다릅니다.

## 2F. 노트 조각

- ✓ N번 정점으로 가는 모든 경로가 최단 경로이므로 최단 경로 조건을 신경쓰지 않아도 됩니다.
- ✓ 이제 남은건 모든 노트 조각을 지난다는 조건이고, 이는 DAG에서 DP로 풀 수 있습니다.
  - $f(x) = \max_{ch \in children[x]} (f(ch) + note[x])$
  - $f(1) < k$  이면  $-1$  출력,  $f(1) = k$  이면 경로 역추적해서 출력
- ✓ 최단 경로 DAG는  $\mathcal{O}(M \log N)$ , DP와 실제 정답은  $\mathcal{O}(N + M)$ 에 구할 수 있습니다.

## 2G. 통행량 조사

trees, graphs, lca

출제진 의도 – Medium

- ✓ 처음 푼 사람:
- ✓ 출제자: SCCC 나정휘

## 2G. 통행량 조사

- ✓  $M = N - 1$  이면 트리입니다.
  - 트리에서 임의의 두 정점을 연결하는 경로는 1가지입니다.
- ✓  $M = N$  이면 트리에 간선 하나를 추가한 그래프입니다.
  - 사이클 하나와 주변에 트리 형태의 가지 여러 개로 구성되어 있습니다.
  - 만약 동일한 트리 안에서 이동한다면 여전히 경로는 1가지입니다.
  - 그렇지 않은 경우, 사이클을 돌아가는 방법이 2가지이므로 경로가 2개 존재합니다.
  - 그러므로 사이클에 속한 간선의 정답은 모두 동일합니다.
- ✓ 사이클과 트리 부분을 잘 분리해야 합니다.

## 2G. 통행량 조사

- ✓ 사이클과 트리 부분을 잘 분리해야 합니다.
  1. 그래프에서 Degree가 1인 정점을 계속 지워나가면, 마지막에 사이클 하나만 남게 됩니다.
  2. 지워지는 정점과 가장 가까운 **사이클 위의 점**을 구하면 트리 부분을 구분할 수 있습니다.
- ✓ 사이클과 트리를 분리하는 것은  $\mathcal{O}(N)$ 에 할 수 있습니다.

## 2G. 통행량 조사

- ✓ 차량 집단의 이동 경로가 주어지면 특정 간선의 가중치를 증가시켜야 합니다.
- ✓ 만약 사이클을 통과한다면 사이클 전부에 대한 정답을 증가시킵니다.
- ✓ 트리 부분의 경우, 트리의 특정 경로 상의 간선의 가중치를 증가시켜야 합니다.
  - HLD과 세그먼트 트리를 사용하면  $\mathcal{O}(\log^2 N)$  에 처리할 수 있습니다.
  - 세그먼트 트리 대신 누적합 배열을 사용하면  $\mathcal{O}(\log N)$  에 처리할 수 있습니다.
  - 오프라인 쿼리라는 점을 활용하면 더 빠르게 해결할 수 있습니다.

## 2G. 통행량 조사

- ✓ 배열의 구간  $[l, r]$ 에  $w$ 를 더하는 것은 누적합을 이용해  $\mathcal{O}(N + M)$ 에 처리할 수 있습니다.
  1.  $S_l$ 에  $w$ 를 더하고
  2.  $S_{r+1}$ 에  $-w$ 를 더하고
  3. 모든 쿼리를 다 처리한 뒤 누적합을 구함
- ✓ 트리의 경로  $(u, v)$  상의 간선에  $w$ 를 더하는 것도 비슷하게  $\mathcal{O}(N + M)$ 에 처리할 수 있습니다.
  1.  $S_u$ 와  $S_v$ 에  $w$ 를 더하고
  2.  $S_{LCA(u,v)}$ 에  $-2w$ 를 더하고
  3. 모든 쿼리를 다 처리한 뒤  $v$ 를 루트로 하는 서브 트리의 가중치 합을  $S_v$ 에 저장
- ✓ 전체 문제를  $\mathcal{O}(N + M)$ 에 해결할 수 있습니다.
- ✓ LCA를 구하는 방법에 따라  $\mathcal{O}(N \log N + M)$ ,  $\mathcal{O}((N + M) \log N)$ 도 가능합니다.



## 2H. 원형 게임

lazyprop

출제진 의도 – **Hard**

- ✓ 처음 푼 사람:
- ✓ 출제자: SCCC 오주원

## 2H. 원형 게임

- ✓ 세그먼트 트리를 이용해 구간의 여러 정보를 관리합니다.
  1. 게임에 참여하는 사람 중 가장 왼쪽 사람의 실력
  2. 게임에 참여하는 사람 중 가장 오른쪽 사람의 실력
  3. 선형 구조에서의 실력 차의 최댓값

## 2H. 원형 게임

- ✓ 3개의 세그먼트 트리를 관리합니다.
  1. 현재 상태
  2. 반전 상태
  3. 모든 인원이 참가할 때의 상태
- ✓ 3번 쿼리에서는 1번 트리와 2번 트리의 정점을 교체합니다.
- ✓ 2번 쿼리에서는 1번 트리에 3번 트리를 덮어씌우고, 2번 트리의 값을 모두 0으로 바꿉니다.
- ✓ 1번 쿼리에서는 2번 쿼리를 적용시킨 다음에 3번 쿼리를 적용시킵니다.

## 2H. 원형 게임

- ✓ 레이저 프로퍼게이션을 사용하면 쿼리당  $\mathcal{O}(\log N)$  의 시간이 걸립니다.
- ✓ 업데이트가 끝난 1번 트리의 1번 노드의 값을 이용해  $\mathcal{O}(1)$  에 답을 구할 수 있습니다.
- ✓ 총 시간복잡도  $\mathcal{O}(M \log N)$  으로 문제를 해결할 수 있습니다.

# 1A. 단어 마방진

implementation

출제진 의도 – **Easy**

- ✓ 처음 푼 사람: **이성현**, 6분
- ✓ 출제자: ALOHA 방인규

## 1A. 단어 마방진

- ✓ 다음과 같이 풀이할 수 있습니다.
  1. 단어를 적당히 선택하여 행렬을 채운다.
  2. 이 행렬이 단어 마방진인지 확인한다.
- ✓ 제한이 작기 때문에 모든 경우를 탐색해도 좋습니다.
- ✓  $i \leq j$  인 모든  $i, j$  에 대해서 행렬의  $i$  행  $j$  열에 적힌 알파벳과  $j$  행  $i$  열에 적힌 알파벳이 같으면 단어 마방진입니다.

## 1A. 단어 마방진

- ✓  $N$  개의 단어 중 총  $L$  개를 선택하여 나열하는 경우의 수는  $\mathcal{O}\left(\frac{N!}{(N-L)!}\right)$  입니다.
- ✓ 나열한 단어로 만든 행렬이 단어 마방진인지는 위 처럼  $\mathcal{O}(L^2)$  에 확인할 수 있습니다.
- ✓ 따라서 완전 탐색을 통해  $\mathcal{O}\left(\frac{L^2 N!}{(N-L)!}\right)$  의 시간복잡도로 해결 할 수 있습니다.

## 1B. 이차 함수

math

출제진 의도 – **Medium**

- ✓ 처음 푼 사람: **최준익**, 18분
- ✓ 출제자: ALPS 노세윤



## 1B. 이차 함수

- ✓  $n = 2$  일 때는 삼각형의 밑변의 길이가 일정하기 때문에 높이를 가장 크게 만들기 위해서는 이등분을 해야합니다.
- ✓  $n = 3$  일 때는 미적분 등을 이용해 계산을 해보면 삼등분할 때 넓이가 최대가 됩니다.
- ✓  $n$  등분 했을 때가 답임을 유추할 수 있습니다.

## 1B. 이차 함수

- ✓ 신발끈 공식을 이용해  $\mathcal{O}(n)$ 에 다각형의 넓이를 구할 수 있습니다.
- ✓ 나눗셈은 페르마 소정리를 이용해서  $\mathcal{O}(\log MOD)$ 에 계산할 수 있습니다.
- ✓ 이제 증명을 해봅시다.

## 1B. 이차 함수

- ✓  $-\int_a^b (x-a)(x-b) = \frac{(b-a)^3}{6}$  임이 잘 알려져 있습니다.
- ✓ 따라서 다각형의 넓이는  $\frac{(b-a)^3}{6} - \frac{(x_1-x_0)^3}{6} - \frac{(x_2-x_1)^3}{6} - \dots - \frac{(x_n-x_{n-1})^3}{6}$  입니다.
- ✓  $\frac{(b-a)^3}{6}$  은 상수이므로  $\frac{(x_1-x_0)^3}{6} + \frac{(x_2-x_1)^3}{6} + \dots + \frac{(x_n-x_{n-1})^3}{6}$  을 최소화 하면 됩니다.

## 1B. 이차 함수

- ✓  $y_i = x_i - x_{i-1}$  로 정의합니다.  $y_1 + y_2 + \cdots + y_n = b - a$  입니다.
- ✓  $f(x) = \frac{x^3}{6}$  이라 할 때  $f(y_1) + f(y_2) + \cdots + f(y_n)$  의 최솟값을 구하면 됩니다.
- ✓  $f(x)$  는  $x > 0$  에서 아래로 볼록한 함수이므로 Jensen 부등식에 의해
$$\frac{f(y_1) + f(y_2) + \cdots + f(y_n)}{n} \geq f\left(\frac{y_1 + y_2 + \cdots + y_n}{n}\right) = f\left(\frac{b-a}{n}\right)$$
 입니다.
- ✓ 등호는  $y_1, y_2, \cdots, y_n$  이 모두 같을 때 성립하므로  $n$  등분하는 것이 최적입니다.
- ✓ 시간 복잡도는  $\mathcal{O}(\log MOD)$  입니다.

# 1C. 캐슬 디펜스

prefix\_sum

출제진 의도 - **Medium**

- ✓ 처음 푼 사람: **이종우**, 62분
- ✓ 출제자: ALPS 노세윤

## 1C. 캐슬 디펜스

- ✓  $t$ 는 클수록 유리하고  $k$ 는 작을수록 유리합니다.
- ✓  $N \leq t < 10^9$ 인 경우는 항상  $t = 10^9$ 으로 대체 가능하기 때문에 답이 될 수 없습니다.
- ✓ 따라서  $1 \leq t < N$ 인 모든  $t$ 에 대해  $k$ 의 최솟값을 구하면 됩니다.
- ✓ 1이상  $i$ 이하의 좌표에 위치한 적의 수를  $S[i]$ 라고 합시다.
- ✓  $k$ 의 최솟값은  $\max_{i \in \mathbb{N}} \left\lfloor \frac{(S[i \cdot t] - E)}{i} \right\rfloor + 1$ 입니다.
- ✓ 시간복잡도를 계산해봅시다.

### 1C. 캐슬 디펜스

- ✓ 보아야 하는  $i$ 의 범위는 대략  $\left[1, \frac{N}{t}\right]$  정도입니다.
- ✓ 따라서 총 시간복잡도는  $\mathcal{O}\left(\sum_{t=1}^N \frac{N}{t}\right)$ 입니다.
- ✓  $\sum_{i=2}^N \frac{1}{i} \leq \int_1^N \frac{1}{x} dx = \ln N$ 입니다.
- ✓ 따라서  $\mathcal{O}\left(\sum_{t=1}^N \frac{N}{t}\right) = \mathcal{O}(N \log N)$ 입니다.
- ✓ 이제 풀이를 증명해봅시다.

## 1C. 캐슬 디펜스

- ✓  $k$  가  $\max_{i \in \mathbb{N}} \lfloor \frac{(S[i \cdot t] - E)}{i} \rfloor + 1$  이상이어야 성이 파괴되지 않는다는 것을 증명하고  
 $k = \max_{i \in \mathbb{N}} \lfloor \frac{(S[i \cdot t] - E)}{i} \rfloor + 1$  일 때 성이 파괴되지 않는다는 것을 증명하면 됩니다.
- ✓ 첫번째부터 증명합시다.
- ✓ 임의의  $i$  에 대하여 성이 받는 데미지는  $S[i \cdot t] - i \cdot k$  이상입니다.
- ✓ 따라서  $S[i \cdot t] - i \cdot k < E$  이고 이는  $k \geq \max_{i \in \mathbb{N}} \lfloor \frac{(S[i \cdot t] - E)}{i} \rfloor + 1$  와 동치입니다.



## 1C. 캐슬 디펜스

- ✓ 두번째를 증명해봅시다.
- ✓ 모든  $i$ 에 대하여  $S[i \cdot t] \leq i \cdot k + E - 1$  입니다.
- ✓ 적들의 수가 같을 때는 위치한 좌표가 작을수록 불리합니다.
- ✓ 따라서 모든  $i$ 에 대하여  $S[i \cdot t] = i \cdot k + E - 1$  인 경우가 가장 불리합니다.
- ✓ 이 경우에 성이 파괴되지 않음은 쉽게 알 수 있습니다.
- ✓ 따라서 증명이 완료되었습니다.
- ✓ 이분탐색을 이용해  $\mathcal{O}(N \log N \log X)$ 로 푸는 방법도 있습니다.

# 1D. 내적

cht

출제진 의도 – Medium

- ✓ 처음 푼 사람: **최준익**, 52분
- ✓ 출제자: ALPS 노세윤

## 1D. 내적

- ✓  $i$  번째 벡터를  $(x_i, y_i)$  라 합시다.
- ✓  $\max_{1 \leq j < i \leq N} x_i x_j + y_i y_j$  를 구해야 합니다.
- ✓ 모든  $i$  에 대해  $\max_{1 \leq j < i} x_i x_j + y_i y_j$  를 빠르게 구할 수 있으면 됩니다.
- ✓  $x_i x_j + y_i y_j = x_i (x_j + y_j \frac{y_i}{x_i})$  입니다.
- ✓  $f_j(x) = x_j + y_j x$  라 하면  $x_i \max_{1 \leq j < i} f_j(\frac{y_i}{x_i})$  를 빠르게 구하는 문제와 같습니다.

## 1D. 내적

- ✓ 두가지 연산을 지원하는 자료구조를 만들면 됩니다.
  - 직선 집합에 직선을 빠르게 추가
  - 주어진  $x$  좌표에서의 함수값의 최댓값을 빠르게 계산
- ✓ 이것은 Convex Hull Trick 이라는 알고리즘으로 해결 가능합니다.
- ✓ 시간복잡도는  $\mathcal{O}(N \log N)$  입니다.
- ✓ 좌표범위가 작기 때문에 CHT에 존재할 수 있는 직선의 최대 개수는  $\mathcal{O}(\sqrt{300\,000})$  입니다.
- ✓ 따라서  $\mathcal{O}(N\sqrt{300\,000})$  으로도 풀 수 있습니다.

# 1E. 다트

convex\_hull

출제진 의도 – Medium

- ✓ 처음 푼 사람:
- ✓ 출제자: ALOHA 이현빈

## 1E. 다트

- ✓ 두 사람이 던진 다트들과 과녁의 위치 관계에 따라 세 가지 경우로 나누어집니다.
  1. 과녁 안에 다트가 명중한 경우, 점수에 과녁의 넓이를 더합니다.
  2. 과녁 가장자리에 다트가 명중한 경우 점수를 얻지 못합니다.
  3. 과녁 밖에 다트가 명중한 경우, 다트에서 과녁에 접선을 그립니다.
- ✓ 따라서 다트와 과녁의 위치 관계를 구해야 합니다.
- ✓  $N, M$  이 최대 50 000 이므로, 각 다트과 과녁의 위치 관계를  $\mathcal{O}(N)$  보다 빠르게 구해야 합니다.
- ✓ 볼록 다각형의 성질을 이용해 이분 탐색을 사용하면  $\mathcal{O}(\log N)$  에 구할 수 있습니다.

## 1E. 다트

- ✓ 다트가 과녁 안에 있는 경우, 미리 구한 전체 과녁의 넓이를 점수에 더합니다.
- ✓ 다트가 과녁 경계에 있는 경우, 0점이므로 아무 것도 하지 않습니다.
- ✓ 다트가 과녁 밖에 있는 경우, 다트에서 과녁에 접선을 그립니다.
  - 접선도 이분 탐색을 이용해  $\mathcal{O}(\log N)$ 에 구할 수 있습니다.
- ✓ 과녁의 두 접점  $L, R$ 에 대하여, 과녁이 선분  $\overline{LR}$ 에 의해 잘린 넓이를 구해야 합니다.

## 1E. 다트

- ✓ 과녁의 부분 넓이를 빠르게 구하기 위해 넓이 누적합 배열을 사용합니다.
- ✓ 과녁의 한 점  $P$ 에 대해, 과녁을  $P$ 를 한 꼭짓점으로 가지는 삼각형들로 분할합니다.
- ✓ 삼각형들의 넓이를 누적합 배열에 저장합니다.
- ✓ 두 점이 주어질 때마다 누적합 배열을 이용하여 과녁의 부분 넓이를  $\mathcal{O}(1)$ 에 구할 수 있습니다.
- ✓ 따라서 전체 시간복잡도는  $\mathcal{O}(N + M \log N)$ 입니다.



# 1F. 르블랑의 트리 순회

trees

출제진 의도 - Hard

- ✓ 처음 푼 사람: **이종우**, 231분
- ✓ 출제자: SCCC 이성서

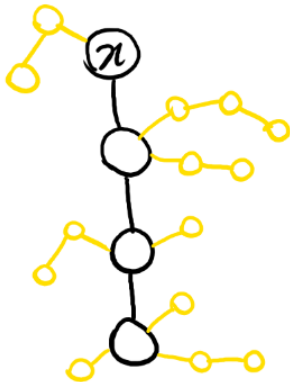
## 1F. 르블랑의 트리 순회

- ✓ 먼저 서브트리의 레벨을 정의합니다.
  - 정점  $x$ 를 루트로 하는 서브트리에 대해,
  - $\text{level}[x] = \max((x\text{의 자식들 중 가장 높은 레벨}), (x\text{의 자식들 중 두 번째로 높은 레벨} + 1))$
  - 처럼 재귀적으로 정의할 수 있습니다.
  - 리프의 레벨은 0으로 하고, 자식이 하나뿐이라면 두 번째 자식에 대한 조건은 무시합니다.

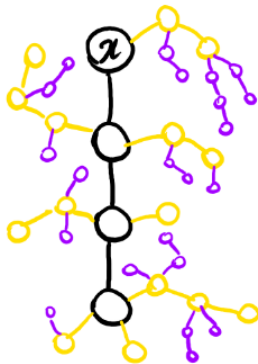
## 1F. 르블랑의 트리 순회



레벨 0 트리



레벨 1 트리



레벨 2 트리

## 1F. 르블랑의 트리 순회

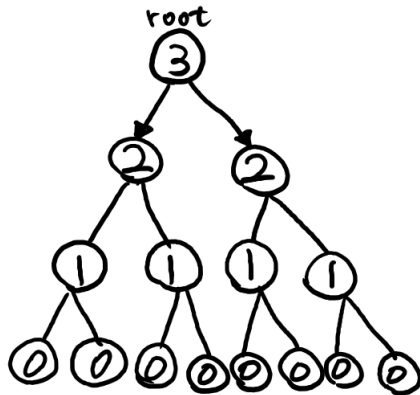
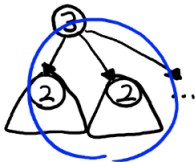
- ✓ 어떤 정점  $s$ 를 루트로 했을 때의 트리 레벨이 2 이하가 된다면,  $s$ 에서 시작하는 적절한 순회가 존재합니다.
  - 구성적으로 증명할 수 있습니다.
  - $s$ 에서부터 DFS를 돌리면서, 자식들 중 레벨이 낮은 서브트리부터 방문합니다.
  - 더 낮은 레벨의 서브트리로 이동할 때마다 체크포인트를 생성하고, 서브트리를 모두 방문한 뒤에 체크포인트로 순간이동해 되돌아옵니다.
  - 항상 2종류 이하의 체크포인트만을 사용하므로 적절한 순회의 조건을 만족합니다.
- ✓ 반면에  $s$ 를 루트로 했을 때의 트리 레벨이 3 이상이 된다면,  $s$ 에서 시작하는 적절한 순회는 존재하지 않습니다.

## 1F. 르블랑의 트리 순회

$\text{level}[x] = \max(\underbrace{(\text{x의 자식들 중 가장 높은 레벨})}, (\text{x의 자식들 중 두 번째로 높은 레벨} + 1))$



$\text{level}[x] = \max(\text{x의 자식들 중 가장 높은 레벨}, \underbrace{(\text{x의 자식들 중 두 번째로 높은 레벨} + 1))$



## 1F. 르블랑의 트리 순회

- ✓ 적절한 순회가 가능한 레벨 3 이상 트리가 존재하는지 판별해 봅시다.
- ✓ 위 페이지의 그림처럼 만드는 게 가장 순회하기 쉽습니다.
- ✓ 하지만 위 트리는 적절한 순회가 불가능하므로 (직관적이지 않다면, 브루트포스 코드를 작성해 확인할 수도 있습니다) 모든 레벨 3 이상 트리는 적절한 순회가 불가능합니다.
- ✓ 이제 전체 문제를 해결할 수 있습니다.

## 1F. 르블랑의 트리 순회

- ✓ 구현 방법 중 하나는 다음과 같습니다.
  1. 차수가 1인 정점들을 모두 찾아서 체크합니다.
  2. 체크된 정점과 인접한, 차수가 2인 정점들을 연속적으로 모두 찾아서 체크합니다.
  3. 체크한 정점들을 모두 제거합니다.
- ✓ 위 과정을 2번 반복한 뒤에 차수가 3 이상인 정점이 없다면 YES, 있다면 NO가 정답입니다.
- ✓ 시간복잡도는  $\mathcal{O}(N)$ 이 됩니다.

# 1G. 숲 게임

game\_theory, linear\_algebra, data\_structures, smaller\_to\_larger

출제진 의도 - Hard

- ✓ 처음 푼 사람:
- ✓ 출제자: SCCC 김도현



## 1G. 숲 게임

- ✓ Sprague-Grundy Theorem에 의해, 그룬디 수를 하나 이상 골라서 XOR한 값이 0이 될 때만 B가 이길 수 있습니다.
- ✓ 일단, 각 트리의 그룬디 수를 계산했다고 가정합시다.
- ✓ basis를 제외한 나머지 벡터(종속 벡터) 집합의 power set을 생각해봅시다.
  - 원소의 합 = 0: 그 자체로 B 승리
  - 그렇지 않은 경우: basis로 덧셈 역원을 만들어서 추가하면 B 승리
- ✓ 종속 벡터 집합의 power set에 대해 모두 0을 만들 수 있으면서
- ✓ basis가 어떤 수 하나를 표현하는 방법은 유일하기 때문에
- ✓ 전체 경우의 수는  $2^{(n-|basis|)} - 1$  입니다.

## 1G. 숲 게임

- ✓ 트리의 그런디 수를 계산해봅시다.
- ✓ 그런디 수는 가능한 다음 상태의 그런디 수 집합의 mex로 정의됩니다.
- ✓ 정점  $x$ 와 거리가  $k$  이하인 정점들의 그런디 수 집합의 mex를 구하면 됩니다.
- ✓ 거리가  $k$  이하인 그런디 수를 `pbds_map<grundy,cnt>` (혹은 세그트리)로 저장했다면 이분탐색을 이용해 mex를  $\mathcal{O}(\log^2 N)$ 에 구할 수 있습니다.
- ✓ small to large를 이용해 두 map을 amortized  $\mathcal{O}(\log^2 N)$ 에 합칠 수 있습니다.
- ✓ 범위를 벗어나는 정점을 제거하기 위해 우선순위 큐도 관리해야 합니다.
- ✓ 이제  $\mathcal{O}(N \log^2 N)$  시간에 문제를 해결할 수 있습니다.

# 1H. DCMSF

matroid, disjoint\_set

출제진 의도 - **Hard**

- ✓ 처음 푼 사람:
- ✓ 출제자: SCCC 나정휘

## 1H. DCMSF

- ✓ 특별한 정점과 멋진 정점 조건이 없다면 그리디하게 해결할 수 있습니다.
- ✓ 포레스트 조건과 멋진 정점 조건이 모두 있어도 그리디하게 해결할 수 있습니다.
- ✓ 위의 두 경우 모두 matroid이기 때문에 그리디하게 최소 가중치 독립 집합을 구할 수 있습니다.

## 1H. DCMSF

- ✓ 특별한 정점 조건이 추가되면 더 이상 matroid가 아닙니다.
- ✓ 하지만 특별한 정점 조건만 있으면 matroid이기 때문에 그리디 알고리즘을 사용할 수 있습니다.
- ✓ 같은 간선 집합에서 정의된 두 matroid의 최소 가중치 공통 독립 집합을 구해야 합니다.
- ✓ 이러한 문제는 matroid intersection이라는 이름으로 잘 알려져 있습니다.
- ✓  $\mathcal{O}(N^2 M \log N)$ 에 해결할 수 있습니다. 이때 log는 Union-Find 연산의 log입니다.