2022 DGIST

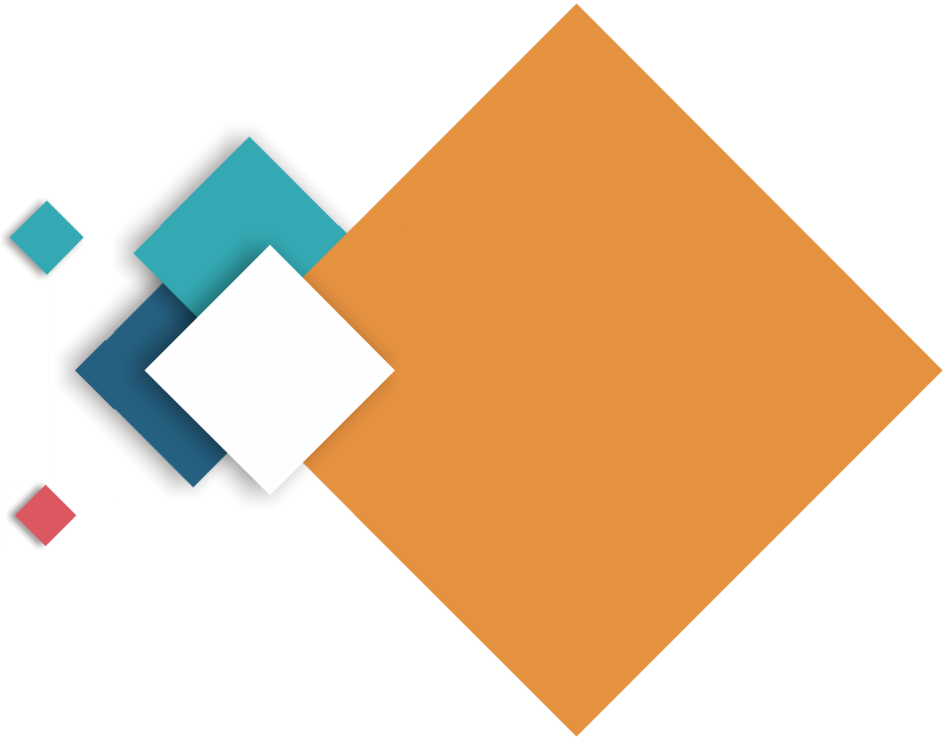
현풍전산배 알고리즘 대회

EDITORIAL

대회 운영진 및 문제 요약

출제진	ekwoo 우은규
	bnb2011 박신욱
검수진	gumgood 이상원
	jhnah917 나정휘
	lky7674 이국렬
	tae826 김태영
	tony9402 김민상

A	에어컨	Prepared by ekwoo
B	비즈마켓	Prepared by bnb2011
C	사각형 게임 (Small)	Prepared by bnb2011
D	사각형 게임 (Large)	Prepared by bnb2011
E	랜선 연결	Prepared by ekwoo
F	뮤직 플레이리스트	Prepared by bnb2011
G	최고의 간선	Prepared by bnb2011
H	천체 관측	Prepared by bnb2011
I	뉴턴의 사과나무	Prepared by bnb2011

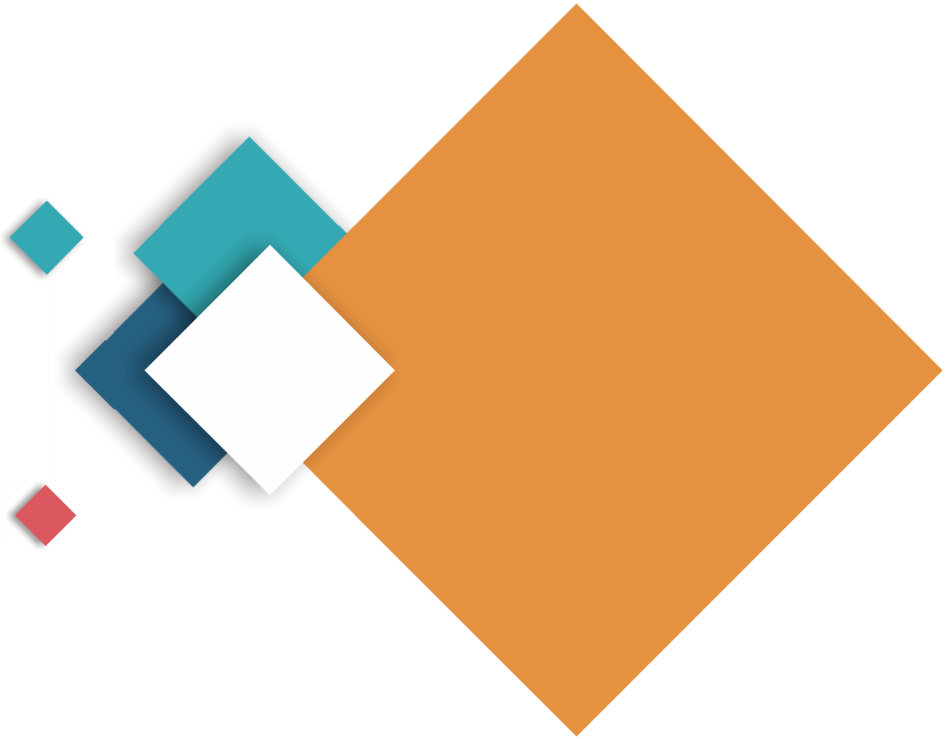


A. 에어컨

출제자: ekwoo
First Solve: 김도훈 (00:09) | wider93 (00:09)
정답률: 16/58 (27.586%) | 49/92 (53.261%)

A. 에어컨 Tag: Implementation, Simulation

주어진 규칙에 따라 특정일에 에어컨이 몇 번 켜지는지 구하는 문제입니다.
에어컨이 켜지는 시각은 $(900 + Mk, 1080 + Mk, 1260 + Mk,\ k$는 0 이상인 정수$)$ 을 만족하는 시각 중 하나이며, N 제한이 10000이기 때문에 가능한 시각의 후보는 최대 3000개입니다. 따라서 답이 될 수 있는 후보 시각들을 전처리 과정에서 다 구해 놓을 수 있습니다.
N일째의 에어컨이 켜지는 시각을 구하기 위해서는 $[1440N,\ 1440N + 1440)$ 구간에 포함되는 시각을 모두 출력하면 됩니다. 시간복잡도는 $O(N)$ 입니다.



B.
비즈마켓

출제자: bnb2011
First Solve: 노지훈 (00:21) | sangyoon9 (00:10)
정답률: 12/61 (19.672%) | 64/194 (32.990%)

B. 비즈마켓 Tag: Greedy, Sorting

$\sum_{i}^{\min(N,M)} \max(A_i - B_i, 0)$ 의 값이 최대가 되도록 수열 A와 B를 잘 배치하는 문제입니다.

아주 직관적으로 생각해봤을 때, A_i의 값은 클수록 이득일 것이고 B_i의 값은 작을수록 이득일 것입니다.

따라서 A를 내림차순으로 정렬하고, B를 오름차순으로 정렬한 뒤 위 식에 따라 계산하면 그게 답이 됩니다.

시간복잡도는 $O(NlogN + MlogM)$ 입니다.

B. 비즈마켓 Tag: Greedy, Sorting

A는 내림차순으로, B는 오름차순으로 정렬된 상태가 최적임을 Exchange Argument를 이용해 증명할 수 있습니다.

$A_{i+1} \geq B_{i+1}$ 이거나 $A_i \leq B_i$ 일 때는 B_i 와 B_{i+1} 을 swap 하더라도 결과값에는 영향을 미치지 않습니다.

$A_i \geq B_i$ 이고 $A_{i+1} \leq B_{i+1}$ 일 때 B_i 와 B_{i+1} 을 swap 하게 되면 결과값이 $A_i - B_i$ 에서 $\max(A_i - B_{i+1}, 0) + \max(A_{i+1} - B_i, 0)$ 으로 변합니다. 결과값이 swap 전보다 항상 같거나 작다면, 주어진 답이 최적임을 증명할 수 있을 것입니다. 대소 관계에 따라 총 4가지 경우가 생기게 됩니다.

1. $A_i \geq B_{i+1}\ and\ A_{i+1} \geq B_i$: $A_i - B_i \geq A_i - B_{i+1} + A_{i+1} - B_i$ 인데 $A_{i+1} \leq B_{i+1}$ 이므로 부등식을 만족합니다.
2. $A_i \geq B_{i+1}\ and\ A_{i+1} \leq B_i$: $A_i - B_i \geq A_i - B_{i+1}$ 인데 $\mathrm{B_i} \leq B_{i+1}$ 이므로 부등식을 만족합니다.
3. $A_i \leq B_{i+1}\ and\ A_{i+1} \geq B_i$: $A_i - B_i \geq A_{i+1} - B_i$ 인데 $A_\mathrm{i} \geq A_{i+1}$ 이므로 부등식을 만족합니다.
4. $A_i \leq B_{i+1}\ and\ A_{i+1} \leq B_i$: $A_i - B_i \geq 0$ 이므로 부등식을 만족합니다.

따라서 A를 내림차순으로, B를 오름차순으로 정렬하는 것이 최적입니다.

C.
사각형 게임 (Small)

출제자: bnb2011
First Solve: 노지훈 (00:38) | aeren (00:18)
정답률: 6/18 (33.333%) | 19/37 (51.351%)

C. 사각형 게임 (Small) Tag: Bruteforcing, Bitmask

N 제한이 9로 매우 작습니다. 따라서 민우가 행을 고르는 2^N가지 경우를 모두 시도해보려는 생각을 할 수 있습니다. 각 경우에 대해서, 종진이가 열을 고르는 경우의 수도 2^N가지가 있습니다. $2^{18} = 1048576$ 이므로 열을 고르는 경우도 모두 시도해볼 수 있습니다.

민우의 점수와 종진이의 점수를 더한 값은 항상 격자에 쓰여진 수들의 합과 같습니다. 따라서 종진이의 최적 전략은 가능한 경우 중 자신의 점수가 가장 높아지도록 열을 고르는 것이고, 민우의 최적 전략은 종진이의 최적 점수가 가장 낮아지도록 행을 고르는 것임을 알 수 있습니다. 이 점만 유의하면 점수를 큰 무리 없이 계산할 수 있습니다.

각 경우에 대해 점수를 계산하는 과정에는 $O(N^2)$ 이 걸리므로, 총 시간복잡도는 $O(N^2 2^{2N})$ 이 됩니다.



D.
사각형 게임 (Large)

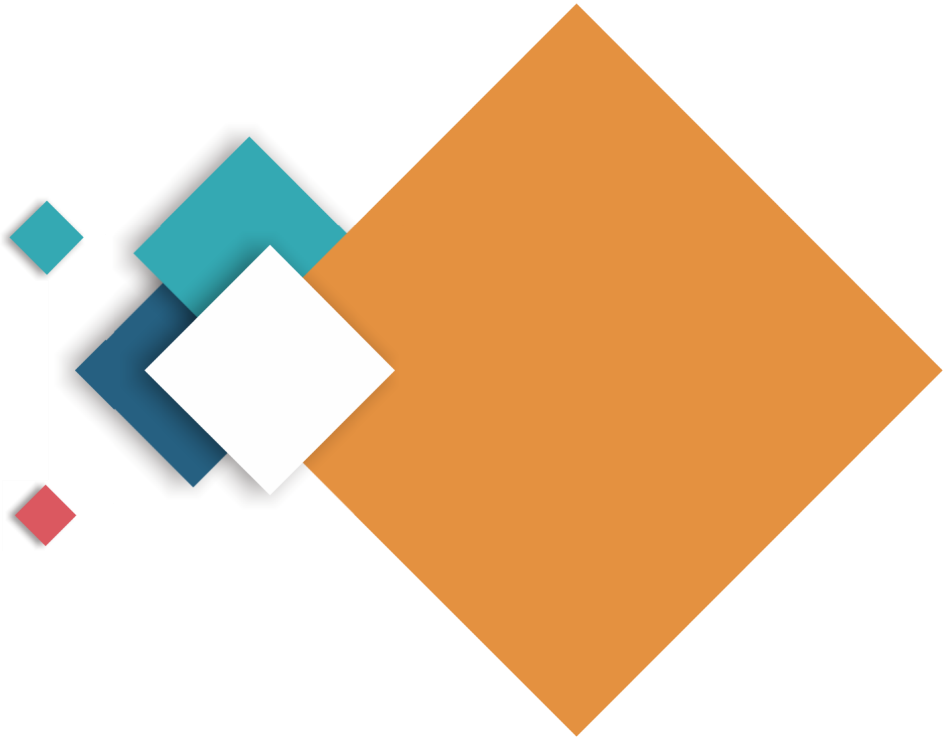
출제자: bnb2011
First Solve: 노지훈 (00:37) | leo020630 (00:24)
정답률: 4/19 (21.053%) | 17/31 (54.839%)

D. 사각형 게임 (Large) Tag: Bruteforcing, Bitmask

N 제한이 18로 조금 커졌습니다. 앞선 풀이를 그대로 제출하게 되면 여기서는 TLE를 받게 됩니다. 그래도 여전히 N이 작긴 하니까, 민우가 행을 고르는 2^N가지 경우를 모두 시도해보겠다는 아이디어는 그대로 유지해 봅시다.

각 열에 대해서, 종진이가 어떤 열을 선택했을 때 얻을 수 있는 점수는 그 열에서 민우가 선택한 행에 적힌 숫자의 합이고, 선택하지 않았을 때 얻을 수 있는 점수는 그 열에서 민우가 선택하지 않은 행에 적힌 숫자의 합입니다. 두 값을 잘 비교해서 종진이가 그 열을 선택할지 말지를 $O(N)$에 판단할 수 있습니다.

총 시간복잡도는 $O(N^2 2^N)$ 입니다.



E.

랜선 연결

출제자: ekwoo
First Solve: 김도훈 (01:53) | sslktong (00:36)
정답률: 2/43 (4.651%) | 11/92 (11.957%)

E. 랜선 연결 Tag: DP, Knapsack

선을 하나 받아와서 $a_i - 1$개의 선을 뿌려주는 스위치를 적절히 사용하여 남는 포트 없이 모든 컴퓨터를 연결해야 하는 문제입니다. $dp[k]$를 k개의 컴퓨터를 남는 포트 없이 연결할 수 있을 때의 최소비용이라고 정의하면, 아래와 같이 점화식을 세워서 해결할 수 있습니다.

$$dp[k] = \begin{cases} 0 & for\ k = 1 \\ \max(dp[k], dp[k - A_i - 1] + B_i) & for\ k \neq 1 \end{cases}$$

랜선이 기본적으로 하나 주어지므로, 모든 스위치를 사용하지 않고 하나의 컴퓨터에 바로 연결하는 경우를 유의해야 합니다. 총 시간복잡도는 $O(NM)$입니다.



F.

뮤직 플레이리스트

출제자: bnb2011
First Solve: 김도훈 (02:52) | snrnsidy (00:36)
정답률: 1/25 (4.000%) | 14/41 (34.146%)

F. 뮤직 플레이리스트 Tag: DP

Sol 1.

잘 생각해보면, 구하고자 하는 값은 전체 수열의 합 + 겹치지 않는 두 부분합의 최대값임을 알 수 있습니다.

겹치지 않는다는 조건에 주목해보면, 주어진 수열을 $[1, i]$ 와 $[\mathrm{i}+1, \mathrm{N}]$ 의 두 부분으로 나눴을 때 각 부분에서의 최대 부분합을 더한 것이 답의 후보가 됩니다.

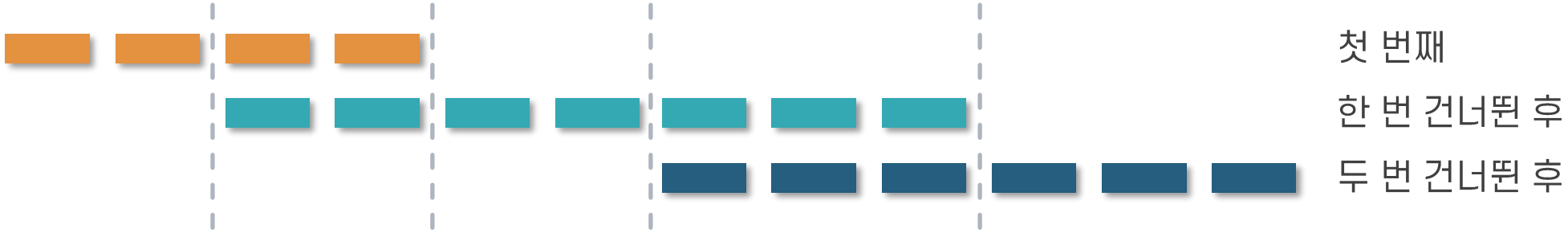
정방향으로 한 번, 역방향으로 한 번 Kadane's Algorithm 을 사용하면 $[1, i]$ 에 해당하는 N개의 최대 부분합과 $[\mathrm{i}+1, \mathrm{N}]$ 에 해당하는 N개의 최대 부분합을 구할 수 있습니다. 이제 구간을 나눌 수 있는 N가지 경우를 모두 시도해보면서 그 중 최대값을 이용해 답을 구하면 됩니다.

총 시간복잡도는 $O(N)$ 입니다.

F. 뮤직 플레이리스트 Tag: DP

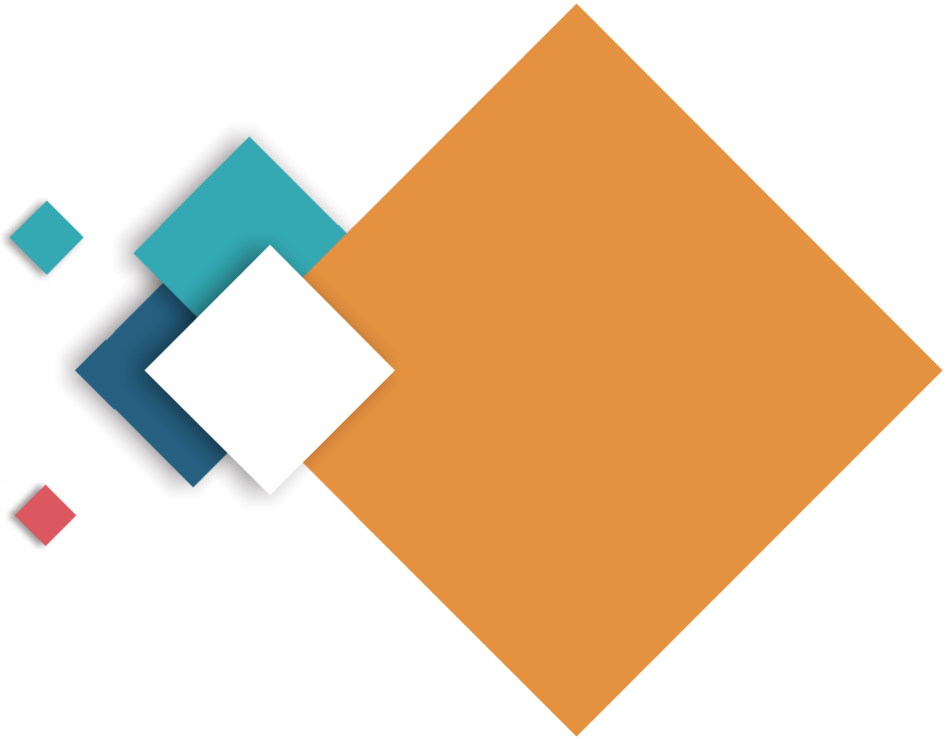
Sol 2.

아래 그림과 같이 현재 노래를 몇 번, 그리고 언제 반복해서 듣느냐에 따라 다섯개의 상태를 정의합니다.



$dp[n][s]$ 를 현재 n번째 노래를 듣고 있고, 상태가 s일 때의 최대 만족도로 정의하면, 다음과 같이 점화식을 세워서 계산할 수 있습니다. 시간복잡도는 Sol. 1과 동일하게 $O(N)$ 입니다.

$$dp[n][s] = \begin{cases} dp[n-1][s] + P_i & for\ s = 0 \\ \max(dp[n-1][s-1], dp[n-1][s]) + 2 \times P_i & for\ s = 1, 3 \\ \max(dp[n-1][s-1], dp[n-1][s]) + P_i & for\ s = 2, 4 \end{cases}$$



G. 최고의 간선

출제자: bnb2011
First Solve: - | aeren (01:00)
정답률: - | 7/23 (30.435%)

G. 최고의 간선 Tag: Dijkstra, DP

Sol 1 (TLE).

처음으로 생각해볼 수 있는 가장 Naive한 방법입니다.

모든 (i,j) 쌍에 대해, Dijkstra를 이용해 i번 정점에서 j번 정점으로 가는 최단 경로를 구합니다. 그리고 경로를 역추적하면서, 최단 경로에 포함된 간선들에 각각 1점씩을 더해줍니다.

(i,j) 쌍은 총 N^2개가 있으므로, Dijkstra와 경로 역추적도 각각 N^2번 해주어야 합니다.

이 풀이의 시간 복잡도는 $O(N^2(N+M)logN)$ 이고, N이 2000이니 TLE를 받게 됩니다.

Sol 2 (TLE).

Dijkstra를 쓰면 i번 정점과 다른 모든 정점 간의 최단 거리를 한꺼번에 구할 수 있으므로, 실제로는 Dijkstra를 N번만 사용해도 됩니다. 그러나 경로 역추적은 여전히 N^2번 해주어야 합니다.

이 풀이의 시간 복잡도는 $O(N(N+M)logN+N^3)$ 이고, N^3 때문에 사이클 형태의 그래프에서 TLE를 받게 됩니다.

G. 최고의 간선 Tag: Dijkstra, DP

Sol 3.

정해에서는 최단 경로로 이루어진 그래프의 특수한 성질을 이용해서 점수를 효율적으로 계산합니다.

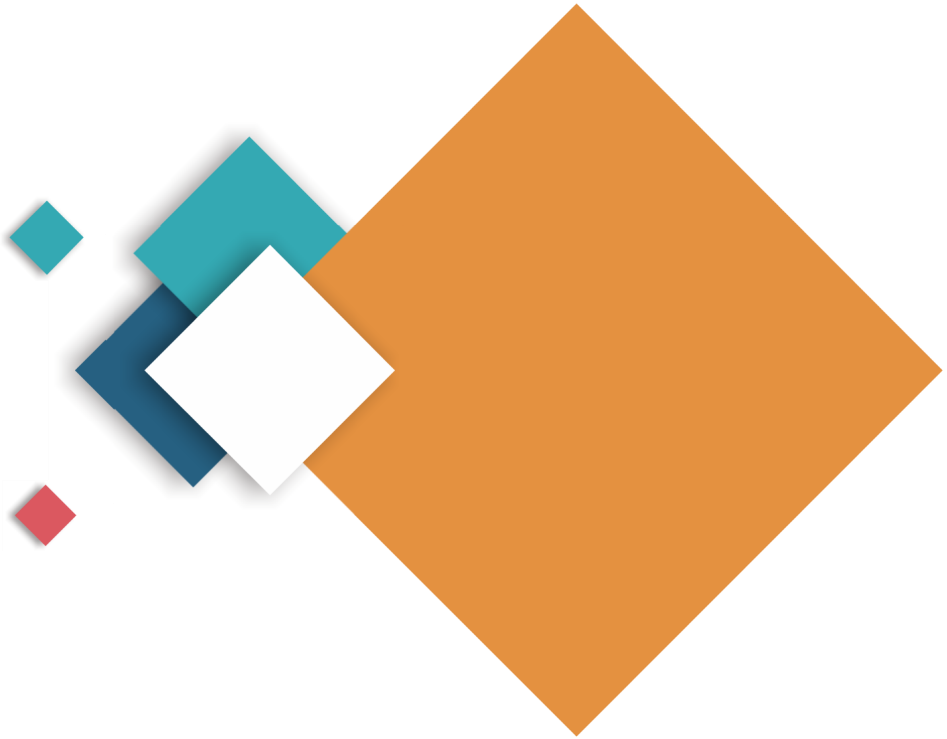
최단 경로에 포함되는 간선들로만 이루어진 그래프는 사이클이 없는 단방향 그래프 (DAG, Directed Acyclic Graph) 이고, 모든 정점쌍의 최단 경로가 유일하다는 조건때문에 DAG 중에서도 Tree의 형태를 가집니다.

Tree의 Root를 Dijkstra의 시작 정점이라고 할 때, 모든 최단 경로는 Root와 나머지 정점을 잇는 경로가 됩니다.

따라서 어떤 간선을 최단 경로가 지나가는 횟수는 그 간선의 도착 정점을 Root로 하는 Subtree의 크기와 같고,

이는 간단한 Tree DP를 이용해 $O(N)$에 계산해줄 수 있습니다.

Dijkstra를 N번, Tree DP를 N번 진행하므로 총 시간복잡도는 $O(N(N+M)logN+N^2)$ 입니다.



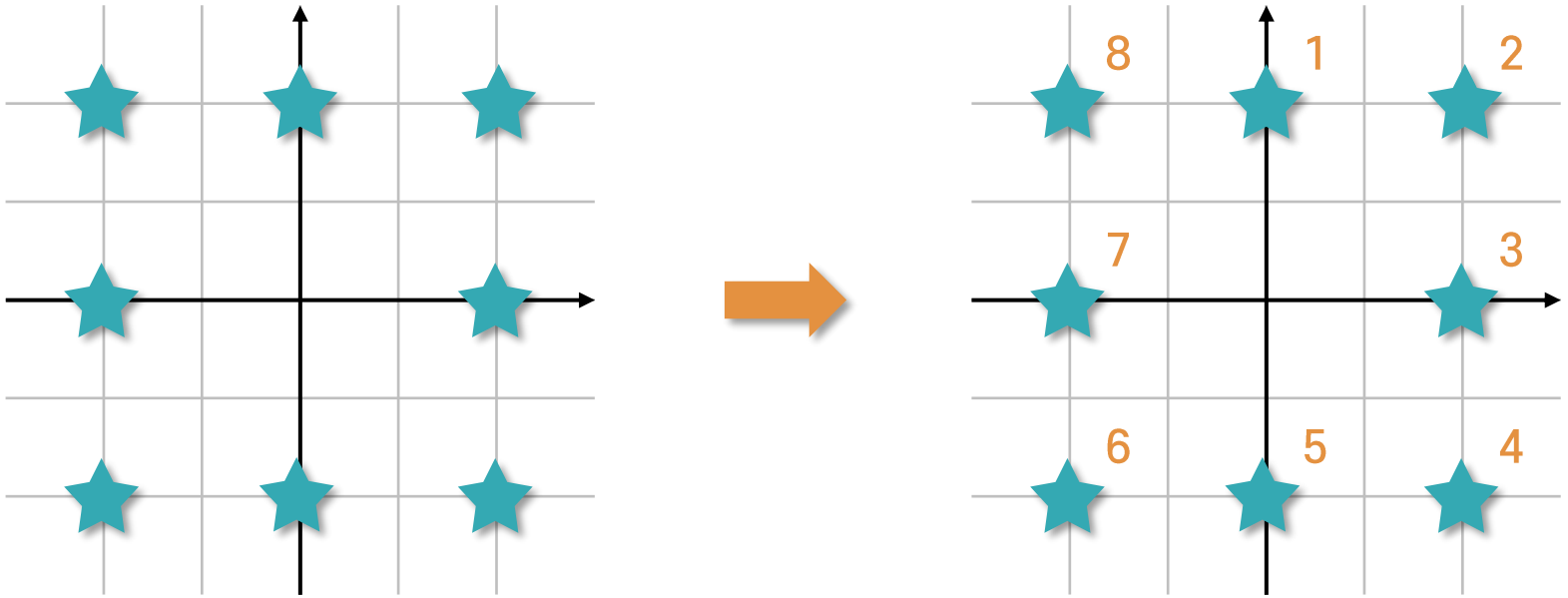
H.

천체 관측

출제자: bnb2011
First Solve: - | aeren (01:16)
정답률: - | 4/14 (28.571%)

H. 천체 관측 Tag: Geometry, Sorting, Two Pointer

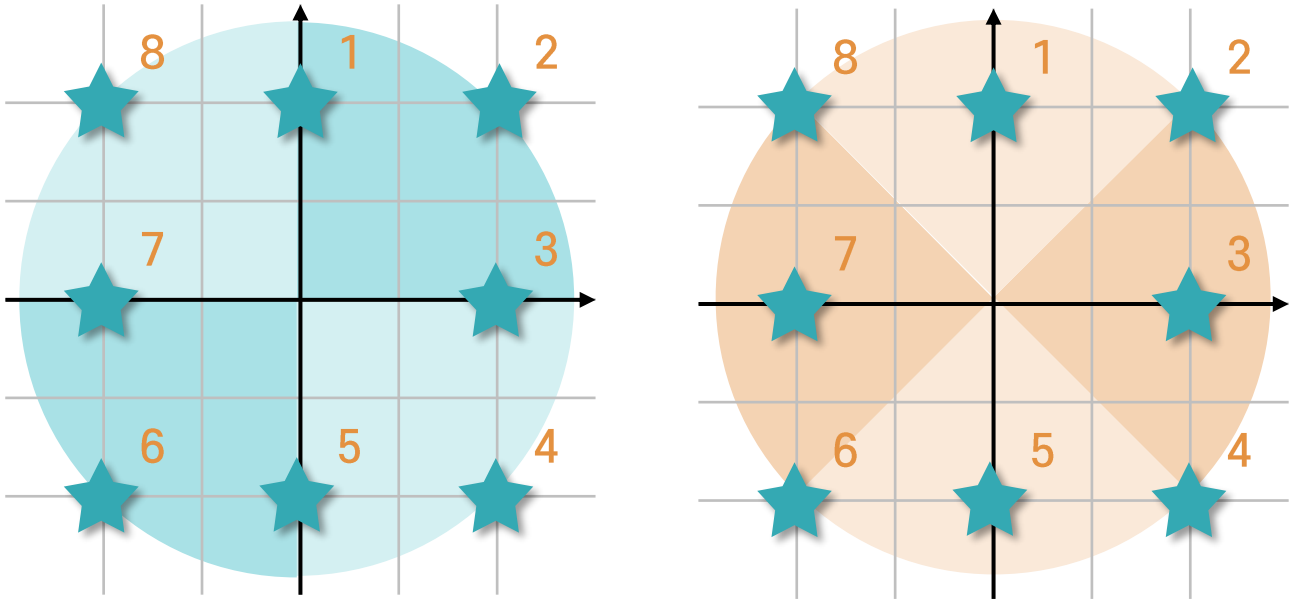
우선 원점을 기준으로 별들을 각도 정렬해줍니다.



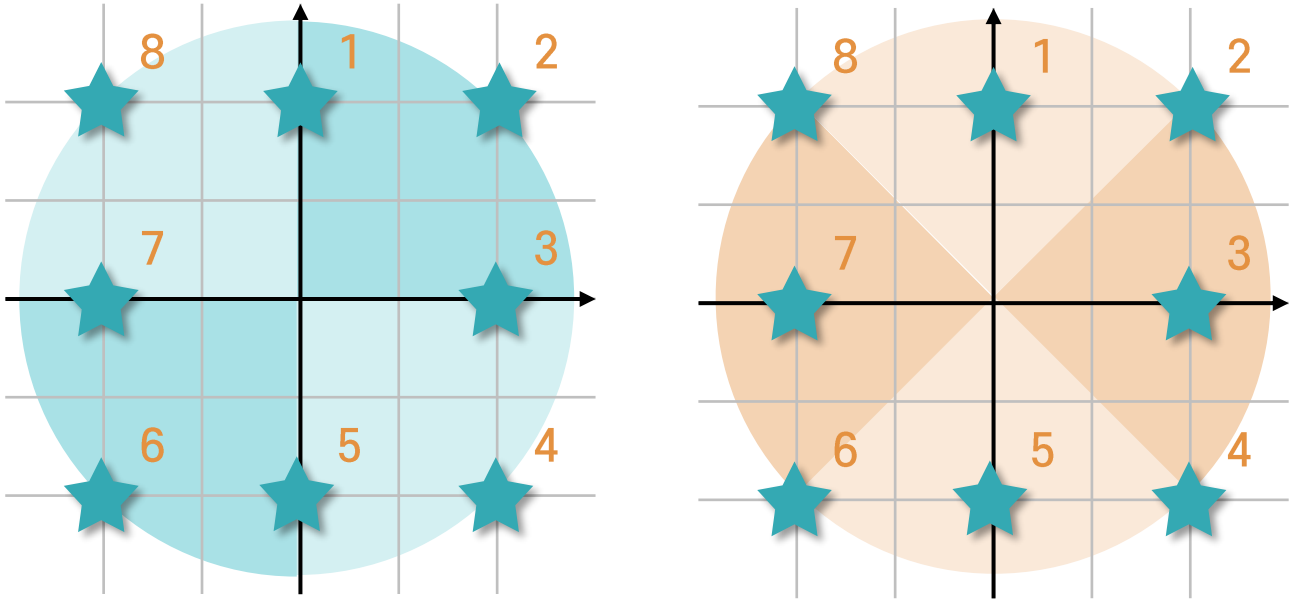
H. 천체 관측 Tag: Geometry, Sorting, Two Pointer

그러고 나서, 가능한 모든 부채꼴의 경우에 대해 별들이 포함되는 구간을 살펴봅니다.

모든 별들의 아름다움은 양수이기 때문에 가능한 많은 별들을 포함하는 것이 최적입니다. 따라서 부채꼴이 어떤 별을 처음으로 포함할 때를 기준으로 두면, 총 N가지의 정답 후보가 나오게 됩니다.



H. 천체 관측 Tag: Geometry, Sorting, Two Pointer



부채꼴이 포함하는 별의 구간이 단조성을 가지고, 내적과 외적의 값이 모두 0 이상이면 어떤 별이 부채꼴 내부에 포함되어 있다고 $O(1)$에 판단 가능합니다. 따라서 Two Pointer를 이용해 가능한 모든 답의 후보를 $O(N)$에 탐색할 수 있습니다.

H. 천체 관측 Tag: Geometry, Sorting, Two Pointer

망원경의 수는 최대 200개이니, 각 망원경에 대해 독립적으로 답을 계산하면 됩니다.
총 시간 복잡도는 $O(NM + NlogN)$ 입니다.



별의 아름다움이 음수가 될 수 있고, 망원경의 개수가 크게 늘어나도 여전히 문제를 해결할 수 있습니다.

I.

뉴턴의 사과나무

출제자: bnb2011
First Solve: - | aeren (01:50)
정답률: - | 2/6 (33.333%)

I. 뉴턴의 사과나무 Tag: Tree, DP, Probability, Math

트리의 루트에서 리프까지 빈틈 없이 색을 쭉쭉 칠해나가는 문제입니다.

Root에서부터 색을 칠해 나가봅시다.
$dp[n][k]$ 를 현재 n번 정점을 n의 $k-1$번째 조상 정점의 색깔으로 색칠했을 때의 기댓값으로 정의합니다.
정의에 따라 $dp[n][1]$ 은 n번 정점을 Root로 하는 Subtree의 기댓값이고, 결과적으로 계산해야 하는 값은 $dp[1][1]$ 이 됩니다.

I. 뉴턴의 사과나무 Tag: Tree, DP, Probability, Math

현재 정점이 Leaf 정점이라면, 더 이상 같은 색을 칠할 수 없으니 (현재 정점부터 $k-1$번째 조상 정점까지의 아름다움의 합) $\times$ k 를 반환합니다. 반환하는 값을 $Psum[n][k]$ 이라고 하겠습니다.
Leaf 정점이 아니라면, 자식 정점들을 균일한 확률로 골라서 같은 색으로 계속 칠해줘야 합니다. 골라지지 않은 자식 정점에서의 기댓값은 $dp[child][1]$ 이고, 고른 정점에서의 기댓값은 $dp[child][k+1]$ 입니다. 따라서 아래와 같이 점화식을 도출해낼 수 있습니다.

$$dp[n][k] = \begin{cases} k \times Psum[n][k] \; for \; k = 1 \\ (n\text{번 정점 자식수의 } Modular\ Inverse) \times \sum_{child}^{childs} \left(dp[child][k+1] - dp[child][1] + \sum_{child}^{childs} dp[child][1] \right) for \; k \neq 1 \end{cases}$$

가능한 상태의 수는 $O(N^2)$ 개가 있고, 각 상태에서의 상태 전이 계산을 평균적으로 $O(1)$ 에 할 수 있습니다. $Psum$ 과 Modular Inverse 값을 전처리해둔 경우에 $O(N^2 + NlogMOD)$에 문제를 해결할 수 있습니다.

I. 뉴턴의 사과나무 Tag: Tree, DP, Probability, Math

C++17	맞았습니다!!	맞았습니다!!	296368 KB	828 ms
C++17	시간 초과	시간 초과		
C++17	틀렸습니다	틀렸습니다		
C++17	맞았습니다!!	맞았습니다!!	198072 KB	1684 ms
C++17	맞았습니다!!	맞았습니다!!	223416 KB	900 ms
C++17	시간 초과	시간 초과		
Java 15	맞았습니다!!	맞았습니다!!	504880 KB	1860 ms
C++17	맞았습니다!!	맞았습니다!!	223508 KB	2756 ms
C++17	맞았습니다!!	맞았습니다!!	296368 KB	1772 ms
C++17	맞았습니다!!	맞았습니다!!	100852 KB	968 ms
C++17	맞았습니다!!	맞았습니다!!	296364 KB	828 ms

C++17	맞았습니다!!	맞았습니다!!	296364 KB	2392 ms
PyPy3	맞았습니다!!	맞았습니다!!	532076 KB	11292 ms
Python 3	시간 초과	시간 초과		
C++17	맞았습니다!!	맞았습니다!!	296684 KB	1584 ms
C++17	맞았습니다!!	맞았습니다!!	296692 KB	3064 ms
C++17	맞았습니다!!	맞았습니다!!	202220 KB	1600 ms
C++17	맞았습니다!!	맞았습니다!!	202300 KB	636 ms
C++17	시간 초과	시간 초과		
C++17	시간 초과	맞았습니다!!	202240 KB	664 ms

출제 과정에서 의도했던 풀이에 따라 제한을 굉장히 넉넉하게 두었습니다. 조금 비효율적이더라도 시간복잡도만 잘 맞췄다면 여유롭게 통과 가능합니다!

! 이 문제는 $O(NlogMOD)$ 에도 해결할 수 있습니다.

Thank You

대회에 참여해주신 모든 분들께 감사드립니다!

