나는코더다 2019 송년대회

경기과학고등학교 35기 나코더

December 12, 2019

대회가 모두 끝났습니다. 수고하셨습니다.

Thanks-to-NEXON Page

문제

출제진 : 윤교준, 김세빈

검수진 : imeimi, jwvg0425, koosaga, rkm0959, TAMREF, cubelover

대회 준비에 참여해주신 분들 모두 감사합니다!

문제

출제진이 생각한 난이도

$$A < D < H < G < B < C < K < L < J < M < I < E < F$$

A. 평면 분할

출제자 : 윤교준

평면 위에 기울기가 -1, 0, 또는 1인 직선을 최대 N개 그릴 때, 분할되는 영역의 수의 최댓값을 구하시오.

A. 평면 분할 - $O(N^2)$

기울기가 -1, 0, 1인 직선이 각각 a, b, c개 있다 합시다.

한 점에서 만나는 세 직선이 없다고 할 때, 분할되는 영역의 수는 $(a+1)(b+1)(c+1) - abc$개입니다.

$a+b+c=N$을 만족하는 모든 (a, b, c) 쌍에 대해 시도해 본 뒤, 최댓값을 출력하면 됩니다.

A. 평면 분할 - $O(1)$

일반성을 잃지 않고 $a \le b \le c$라 합시다.

다음 명제를 증명할 수 있습니다. 증명은 생략합니다.

Lemma

영역의 갯수가 최대가 될 때, $c - a \le 1$이다.

이 사실을 이용하면, $a = \left\lfloor \frac{N}{3} \right\rfloor$, $b = \left\lfloor \frac{N+1}{3} \right\rfloor$, $c = \left\lfloor \frac{N+2}{3} \right\rfloor$일 때 영역의 갯수가 최대가 된다는 것을 알 수 있습니다.

A. 평면 분할

- Tag : 쉬운 문제.
- 출제 및 검수진의 최단 코드 : 200 bytes(C++),
  49 bytes(Python)
- First to Solve : 지학 전공 (4 min)
- Open Contest에는 출제되지 않았습니다.

D. 다오의 데이트

출제자 : 윤교준

다오는 디지니와 만나 데이트를 하고 싶다. 이차원 맵 위에서 각 턴에 정해진 두 방향 중 한 방향으로만 움직일 수 있을 때, 디지니를 만날 수 있는지 판단하고 가능하다면 그 방법을 구하시오.

D. 다오의 데이트 - $O(HW + 2^N)$

한 번에 두 가지 방향 중 하나를 선택할 수 있으므로, 가능한 방법의 수는 최대 2^N입니다.

$N \leq 20$이므로, 모든 방법을 해 보면서 가능한지 판단하면 됩니다.

D. 다오의 데이트

- Tag : 전체 탐색, 구현
- 출제 및 검수진의 최단 코드 : 870 bytes
- First to Solve : 1111 (38 min)
- First to Solve (Open) : povwhm (27 min)

H. 보고 정렬

출제자 : 김세빈

순열에서 어떤 연속한 구간을 잡아 랜덤 셔플하는 동작만을 반복하여 수열을 정렬하시오.

H. 보고 정렬 - $O(N \log N)$

다음 두 가지를 반복하는 알고리즘을 사용하면, 함수를 호출하는 횟수의 기댓값이 대략 $O(N \log N)$이 됩니다.

1. $A_s \neq s$를 만족하는 최소의 s를 찾는다. s가 존재하지 않는다면 수열이 정렬된 것이므로 종료한다.
2. $A_e = s$를 만족하는 e를 찾고, $[s, e]$를 랜덤 셔플한다.

H. 보고 정렬 - $O(N \log N)$

$N = 200$일 때의 기댓값은 대략 1060으로, 2500번 안에 충분히 정렬할 수 있습니다. 실제로 검수 과정에서 함수 호출 횟수가 1200번을 넘은 경우가 한 번도 발생하지 않았습니다.

H. 보고 정렬 - $O(N \log N)$

시간 복잡도를 증명해봅시다! 어떤 수를 n칸 왼쪽으로 옮기는 데 필요한 횟수의 기댓값을 구하면 됩니다.

i칸 옮기는 횟수의 기댓값을 D_i라 하면, 다음이 성립합니다.

$$D_n = \frac{1}{n+1}(D_1 + D_2 + \cdots + D_n) + 1$$

H. 보고 정렬 - $O(N \log N)$

$S_n = D_1 + D_2 + \cdots + D_n$이라 하고, 대입한 뒤 정리하면

$$
\begin{aligned}
S_n &= \frac{n+1}{n}(S_{n-1}+1) \\
&= \frac{n+1}{n} + \frac{n+1}{n}\cdot\frac{n}{n-1} + \frac{n+1}{n}\cdot\frac{n}{n-1}\cdot\frac{n-1}{n-2} + \cdots \\
&= (n+1)\left(\frac{1}{n} + \frac{1}{n-1} + \cdots + 1\right)
\end{aligned}
$$

H. 보고 정렬 - $O(N \log N)$

$D_n = S_n - S_{n-1}$을 계산해봅시다.

$$
\begin{aligned}
D_n &= S_n - S_{n-1} \\
&= 1 + \sum_{i=1}^{n} \frac{1}{i} \\
&\approx 2 + \ln n
\end{aligned}
$$

따라서 총 호출 횟수의 기댓값은 $O(N \log N)$입니다.

H. 보고 정렬

- Tag : Interactive, 정렬, 랜덤 알고리즘.
- 출제 및 검수진의 최단 코드 : 239 bytes
- First to Solve : KR815KR (19 min)
- First to Solve (Open) : kcm1700 (29 min)

G. 피아노 연주

출제자 : 김세빈

M개의 음으로 이루어진 피아노 곡이 주어진다. 각각의 음에 손가락을 잘 배정하여 곡의 난이도를 최소화하시오.

G. 피아노 연주 - $O(M \log X)$

인접한 두 음을 연주할 때의 난이도 $|(P_{i+1} - P_i) - (X_{F_{i+1}} - X_{F_i})|$를 손이 움직이는 거리라고 해석할 수 있습니다.

손이 한 번에 움직이는 최대 거리를 최소화하는 문제가 됩니다.

G. 피아노 연주 - $O(M \log X)$

손이 한 번에 움직일 수 있는 거리를 최대 L로 제한하고, 이 제한을 지키면서 곡을 연주하는 것이 가능한지 판단해 봅시다.
이것이 가능하다면 이분 탐색을 이용하여 가능한 최소 L을 찾을 수 있고, 이 최소 L이 바로 답이 됩니다.

G. 피아노 연주 - $O(M \log X)$

곡의 첫 번째 음은 1번부터 N번까지의 모든 손가락으로 연주할 수 있습니다. 하지만 두 번째 음부터는 그 음을 연주하지 못하는 손가락이 생길 수 있습니다.

이때 손가락들 사이의 거리가 K로 일정하므로, 각 음을 연주할 수 있는 손가락들의 집합은 구간을 이루게 됩니다.

G. 피아노 연주 - $O(M \log X)$

첫 번째 음부터 시작하여 하나씩 넘어가면서 각 음을 연주할 수 있는 손가락들을 나타내는 구간을 구해 봅시다. 만약 어떤 음에서 구간이 존재하지 않는다면, 그 L은 불가능한 것입니다.

이와 같이 거리 제한 L이 가능한지 $O(M)$에 판단할 수 있습니다.

이분 탐색을 고려하면 총 시간복잡도는 $O(M \log X)$입니다.

G. 피아노 연주

- Tag : 이분 탐색
- 출제 및 검수진의 최단 코드 : 711 bytes(C++),
  374 bytes(Python)
- First to Solve : 뚝심햄구이 (214 min)
- First to Solve (Open) : tlwpdus (22 min)

B. 라면 사기 (Small)

출제자 : 윤교준

N개의 라면 공장이 있고, 각각의 공장에서 정해진 만큼 라면을 구입하고 싶다. 이때 3, 5, 7의 비용으로 각각 연속한 1개, 2개, 3개의 공장에서 라면을 한 개씩 구입할 수 있다. 필요한 비용의 최솟값을 구하시오.

B. 라면 사기 (Small) - $O(N)$

1번부터 i번 공장까지 최소 비용으로 라면을 구입하는 해를 구했다고 가정합시다. 이를 이용해 $i+1$번 공장까지 고려한 최적해를 만들 수 있습니다.

B. 라면 사기 (Small) - $O(N)$

다음 세 가지를 순서대로 가능한 만큼 시행하면 됩니다.

1. i번 공장에서 비용 3으로 라면을 구입한 것을 늘려 i, $i+1$번 공장에서 비용 5로 라면을 구입하게 한다.
2. $i-1$, i번 공장에서 비용 5으로 라면을 구입한 것을 늘려 $i-1$, i, $i+1$번 공장에서 비용 7로 라면을 구입하게 한다.
3. $i+1$번 공장에서 비용 3으로 라면을 구입한다.

자세한 정당성 증명은 생략합니다.

B. 라면 사기 (Small)

- Tag : 그리디
- 출제 및 검수진의 최단 코드 : 479 bytes
- First to Solve : 킹장배빵빵 (80 min)
- Open Contest에는 출제되지 않았습니다.

B. 라면 사기 (Large)

출제자 : 윤교준

Small에서 N 제한이 10^6으로 늘어났고, 비용이 3, 5, 7에서 B, $B + C$, $B + 2C$로 변경되었습니다.

B. 라면 사기 (Large)

- Tag : 그리디
- 출제 및 검수진의 최단 코드 : 496 bytes
- 교내 대회에는 출제되지 않았습니다.
- First to Solve (Open) : august14 (11 min)

C. 분할하기

출제자 : 윤교준

어떤 집합이 bitwise-or 연산에 대해 닫혀 있다면 그 집합을 좋은 집합이라 한다. 2^N 미만의 음이 아닌 정수로 이루어진 크기 K의 집합 S를 구하시오. 단, S와 S^C는 좋은 집합이어야 한다.

C. 분할하기 - $O(2^N)$

PS의 불문율 : 예제에 NO가 없다 $\Rightarrow$ 답은 항상 YES이다

크기가 각각 K, $N - K$이면서 서로 교집합이 없는 좋은 집합을 찾으면 됩니다.

여러 가지 방법이 가능하겠지만, 가장 간단한 방법을 소개합니다.

C. 분할하기 - $O(2^N)$

2^N개의 수들을 최상위 비트가 같은 것끼리 묶어 봅시다.

$$\{0\}, \{1\}, \{2,3\}, \{4,5,6,7\}, \cdots, \{2^{N-1}, 2^{N-1}+1, \cdots, 2^N - 1\}$$

이 묶음들 중 몇 개를 합집합하여 만든 집합은 좋은 집합이라는 것을 알 수 있습니다!

C. 분할하기 - $O(2^N)$

각 묶음의 크기가 $1, 1, 2, 4, \cdots, 2^{N-1}$이므로, 묶음들을 합집합하여
1 이상 2^N 이하의 모든 크기의 좋은 집합을 만들 수 있습니다.
또한 이렇게 만들어진 집합의 여집합도 자연히 좋은 집합이 됩니다.
결국, 최상위 비트가 K에 포함된 수를 모두 출력하면 답이 됩니다.

C. 분할하기

- Tag : 비트 연산, Constructive, 아이디어
- 출제 및 검수진의 최단 코드 : 239 bytes
- First to Solve : 대가리 큰 곰돌이 푸를 사냥하는 명 윅은 노래를 부르다가 특별상만을 노린다. (26 min)
- First to Solve (Open) : ainta (15 min)

K. 정점 찾기

출제자 : 윤교준

정점 N개로 이루어진 그래프에 간선이 하나 숨겨져 있다. 여러분은 그래프에 간선을 몇 개 추가한 뒤, 어떤 두 정점이 간선을 통해 연결되어 있는지 묻는 질문을 할 수 있다. 숨겨진 간선을 찾으시오.

K. 정점 찾기

질문 13번이 반드시 필요할까요?

그래프에서 가능한 간선의 갯수는 $\frac{N(N-1)}{2}$입니다.

한 번의 질문으로 얻을 수 있는 답은 0과 1 두 가지입니다.

$\log_2 \frac{N(N-1)}{2} \approx 2\log_2 N - 1$이므로, 질문은 최소 $\lceil 2\log_2 N - 1 \rceil$ 회는 필요합니다.

이 문제에서는 $2\lceil \log_2 N \rceil - 1$회의 질문으로 해결하면 됩니다.

K. 정점 찾기 - $O(\log N)$

다음과 같은 질문을 해 봅시다.

- i번째 비트가 0인 정점들끼리 모두 연결하고, i번째 비트가 1인 정점들들끼리 모두 연결하였을 때, 0번 정점과 2^i번 정점이 연결되어 있는가?

질문의 답이 1이라면, i번째 비트가 0인 정점과 1인 정점을 연결하는 간선이 존재한다는 것이기 때문에, 간선이 연결하는 두 정점의 인덱스의 i번째 비트가 다르다는 것을 알 수 있습니다.

K. 정점 찾기 - $O(\log N)$

따라서 이와 같은 질문 $\lceil \log_2 N \rceil$회로 $A \oplus B$ (A와 B의 exclusive-or)의 값을 알아낼 수 있습니다. 이제 남은 질문 $\lceil \log_2 N \rceil - 1$회로 A와 B의 정확한 값을 알아내면 됩니다.

$A \oplus B = X$라 합시다. $p \oplus q = X$를 만족하는 모든 $p < q$에 대해, p를 집합 P에 추가하고, q를 집합 Q에 추가합시다.

K. 정점 찾기 - $O(\log N)$

두 집합 P, Q를 이용한 적절한 이분 탐색을 통해 A와 B의 정확한 값을 찾을 수 있습니다.
집합의 크기는 최대 $\frac{N}{2}$이므로, 질문 $\lceil \log_2 N \rceil - 1$회로 충분합니다.

K. 정점 찾기

- Tag : Interactive, 이분 탐색
- 출제 및 검수진의 최단 코드 : 907 bytes
- First to Solve : KR815KR (165 min)
- First to Solve (Open) : august14 (99 min)

L. 정기 모임

출제자 : 김세빈

N개의 정점으로 이루어진 가중치 있는 트리가 주어진다. 다음 쿼리를 처리하여라 : 정점 v를 잡아 구간 $[S_i, E_i]$의 각 정점들과 v를 잇는 최단 경로 상의 간선들의 가중치들 중 최댓값을 최소화하시오.

L. 정기 모임 - $O((N+Q)\log N)$

정점 v가 구간 $[S_i, E_i]$에 속할 때가 최적입니다.

결국, 주어진 트리의 간선들을 사용해 구간 $[S_i, E_i]$의 정점들을 모두 연결하는 최소의 트리에서 간선들의 가중치의 최댓값을 구하는 문제가 됩니다.

L. 정기 모임 - $O((N+Q)\log N)$

정점 S_i와 S_i+1을 연결하는 최단경로, 정점 S_i+1과 S_i+2를 연결하는 최단경로, ... , 정점 E_i-1과 E_i를 연결하는 최단경로를 사용하여 $[S_i, E_i]$의 정점들을 연결하는 최소의 트리를 정확히 커버할 수 있습니다.

정점 i와 $i+1$을 잇는 최단경로 상의 가중치의 최댓값을 X_i라 합시다. 이때 쿼리의 답은 $X_{S_i}, X_{S_i+1}, \cdots, X_{E_i}$ 중 최댓값이 됩니다.

L. 정기 모임 - $O((N+Q)\log N)$

X의 값은 sparse table을 사용하여 $O(N\log N)$에 구할 수 있습니다.
쿼리에 대한 답은 세그먼트 트리를 사용하여 쿼리당 $O(\log N)$에 할 수 있습니다.

L. 정기 모임 - $O((N+Q)\log N)$, 별해

주어진 트리와 거리 관계가 동일한 선형 트리를 만들 수 있습니다. 선형 트리에서 인덱스를 순서대로 새로 매긴 뒤, $[S_i, E_i]$에 속하는 정점들 중 새로 매겨진 인덱스가 가장 작은 정점과 큰 정점을 구합니다. 구한 두 정점 사이의 간선의 가중치들 중 최댓값을 출력하면 됩니다.

L. 정기 모임

- Tag : 트리, 자료 구조
- 출제 및 검수진의 최단 코드 : 1661 bytes
- First to Solve : 뚝심햄구이 (195 min)
- First to Solve (Open) : onjo0127 (158 min)

J. Bad Hair Day와 기댓값

출제자 : 김세빈

N마리의 소들을 임의로 일렬로 세울 때, 머리스타일을 확인할 수 있는 소들의 쌍의 수의 기댓값을 구하시오.

J. Bad Hair Day와 기댓값 - $O(N \log N)$

각각의 소들에 대해, 그 소가 머리스타일을 확인할 수 있는 소들의 수의 기댓값을 구한 뒤 이를 모두 합하면 됩니다.

i번째 소가 j번째 소를 볼 수 있을 확률을 구해 봅시다. i번째 소보다 키가 작은 소의 수를 k라 합시다.

먼저 $H_i \leq H_j$일 경우, 확률은 0입니다.

J. Bad Hair Day와 기댓값 - $O(N \log N)$

$H_i > H_j$일 경우, i번째 소와 j번째 소 사이에 키가 H_i보다 작은 소들만이 올 확률을 구하면 됩니다. 다음과 같은 세 단계를 통해 원하는 상황을 얻을 수 있습니다.

1. 키가 H_i 이상인 소들을 아무렇게나 배치한다.
2. j번째 소를 i번째 소 바로 오른쪽에 끼워 넣는다.
3. 나머지 $k-1$마리의 소들을 아무렇게나 끼워 넣는다.

J. Bad Hair Day와 기댓값 - $O(N \log N)$

세 단계 중 확률에 영향을 미치는 것은 두 번째 단계밖에 없습니다!

j번째 소를 끼워 넣을 수 있는 공간이 $N - k + 1$개 있으므로,

확률은 $\frac{1}{N-k+1}$이 됩니다. 따라서 i번째 소가 머리스타일을

확인할 수 있는 소들의 수의 기댓값은 $\frac{k}{N-k+1}$입니다.

따라서 모든 소들에 대해 자신보다 키가 작은 소들의 수를 센 뒤,

위의 식을 계산하여 모두 더하면 답이 됩니다.

J. Bad Hair Day와 기댓값 - $O(N \log N)$

좀 더 복잡한 방법으로도 답을 구할 수 있습니다.

i번째 소가 확인할 수 있는 소들이 정확히 j마리일 확률은

$$\frac{(N-k)k!(N-j-1)!}{(k-j)!N!}$$

따라서 기댓값은

$$\frac{(N-k)k!}{N!}\sum_{j=0}^{k}\frac{(N-j-1)!}{(k-j)!}j \tag{1}$$

J. Bad Hair Day와 기댓값 - $O(N \log N)$

텔레스코핑 등의 방법으로 식 (1)의 값을 구하면 동일한 결과를 얻습니다. 자세한 과정은 너무 길어 생략합니다.

J. Bad Hair Day와 기댓값

- Tag : 수학, 조합론
- 출제 및 검수진의 최단 코드 : 519 bytes(C++),
  249 bytes(Python)
- First to Solve : 뚝심햄구이 (136 min)
- First to Solve (Open) : ainta (159 min)

M. 분수 계산

출제자 : 윤교준

0과 1 사이 유리수로 이루어진 수열이 주어졌을 때, 그 수열의 분포 정도가 같은 길이를 갖는 수열들 중 최대인지 판단하시오.

M. 분수 계산 - $O(N)$

거리 함수의 정의가 조금 특이합니다.

원주의 길이가 1인 원을 그리고, 원주 위에 원점을 잡아 0에서 1까지의 실수를 원주 위의 각 점에 대응시킵시다. 이때 두 수의 거리는 원주 위에서의 두 점의 거리가 됩니다.

M. 분수 계산 - $O(N)$

분포 정도가 최대일 때, 다음 명제가 성립합니다.

Lemma

원 위에 찍힌 점들을 통과하지 않는 임의의 지름을 잡았을 때, 지름의 양쪽에 찍혀 있는 점들의 갯수의 차이는 1 이하이다.

만약 점들의 갯수가 1 이상 차이난다면, 점들이 많은 쪽에서 지름에 가장 가까운 점 하나를 골라 지름 쪽으로 살짝 이동시켰을 때 분포 정도가 더 커지게 됩니다.

M. 분수 계산 - $O(N)$

앞의 명제가 분포 정도가 최대이기 위한 필요충분조건이라는 사실 또한 증명할 수 있습니다. 자세한 과정은 생략합니다.

따라서 주어진 수열이 명제의 조건을 만족하는지 검사하면 됩니다.

M. 분수 계산 - $O(N)$

N이 짝수일 때와 홀수일 때를 나누면 조금 편합니다.

편의상 $A_{N+i} = A_i$라 합시다. 아래 두 조건을 만족하면 `NO`입니다.

- N이 짝수일 때, A_i의 대칭점은 $A_{\frac{N}{2}+i}$와 같아야 합니다.
- N이 홀수일 때, A_i의 대칭점은 $A_{\frac{N-1}{2}+i}$와 $A_{\frac{N+1}{2}+i}$ 사이에 있어야 합니다.

M. 분수 계산 - $O(N)$

주어지는 수들이 모두 유리수이기 때문에, 큰 소수 p를 잡아 분포 정도를 해싱하여 $A_i = \frac{i}{N}$인 수열과 비교하는 방법도 있습니다. 정답으로 인정합니다. 사실 막을 수가 없어요.

M. 분수 계산

- Tag : 수학
- 출제 및 검수진의 최단 코드 : 615 bytes
- First to Solve : 의문부호의 꼭두각시 (152 min)
- First to Solve (Open) : ainta (90 min)

I. 비행기 타고 가요

출제자 : 윤교준

N개의 정점으로 이루어진 그래프가 있다. 또한, 구간 $[A_i, B_i]$의 각 정점에서 출발하여 구간 $[C_i, D_i]$의 각 정점에 도착하는 가중치 E_i의 간선들이 있다. K번 정점에서 각 정점까지 가는 최단 경로의 길이를 구하시오.

I. 비행기 타고 가요 - $O(N \log N + M \log^2 N)$

간선이 너무 많이 생깁니다. 각 정점들 사이의 거리 관계를 유지하면서 간선의 갯수를 줄이는 방법을 생각해 봅시다.

I. 비행기 타고 가요 - $O(N \log N + M \log^2 N)$

먼저, 다음과 같은 방법을 생각할 수 있습니다.

- 더미 노드 u_i를 추가하고, 구간 $[A_i, B_i]$의 각 정점에서 u_i로 가는 가중치 0인 간선들을 추가한다.
- 더미 노드 v_i를 추가하고, v_i에서 구간 $[C_i, D_i]$의 각 정점으로 가는 가중치 0인 간선들을 추가한다.
- u_i에서 v_i로 가는 가중치 E_i인 간선을 추가한다.

하지만 여전히 간선이 너무 많습니다.

I. 비행기 타고 가요 - $O(N \log N + M \log^2 N)$

구간의 각 정점들을 더미 노드와 연결하는 과정을 구간 쿼리로 생각할 수 있지 않을까요?

세그먼트 트리로 구간 쿼리를 빠르게 해결하는 것처럼, 세그먼트 트리를 사용하여 간선의 갯수를 줄여 봅시다.

구간 $[s, e]$를 나타내는 정점과 정점 v를 간선으로 연결하면, 그 구간에 속한 각 정점들과 v를 직접 연결한 것과 동일한 효과가 나타나야 합니다.

I. 비행기 타고 가요 - $O(N \log N + M \log^2 N)$

다음과 같이 세그먼트 트리 형태로 그래프를 그려 봅시다.

1. 구간 $[1, N]$을 나타내는 정점을 만든다.
2. 구간 $[s, e]$를 나타내는 정점이 있을 때, m을 중점이라 하자.
   1. 구간 $[s, m]$을 나타내는 정점을 만들고, $[s, m]$에서 $[s, e]$로 가는 가중치 0의 간선을 추가한다.
   2. 구간 $[m + 1, e]$을 나타내는 정점을 만들고, $[m + 1, e]$에서 $[s, e]$로 가는 가중치 0의 간선을 추가한다.

I. 비행기 타고 가요 - $O(N \log N + M \log^2 N)$

이제 구간 $[A_i, B_i]$에서 정점 u로 가는 간선을 만들고 싶을 때, 합집합하여 $[A_i, B_i]$를 만들 수 있는 $\log N$개의 구간을 골라 각 구간을 나타내는 정점들에서 u로 가는 간선을 추가하면 됩니다.

v에 대해서도 마찬가지로 트리를 하나 더 만들면 됩니다.

이렇게 하면 간선이 총 $O(N + M \log N)$개 만들어집니다. 여기에서 다익스트라 알고리즘으로 최단경로를 구하면 됩니다.

I. 비행기 타고 가요 - $O(N\log^2 N + M\log N)$

세그먼트 트리 대신 sparse table처럼 그래프를 만들면 두 개의 구간만 합집합하여도 $[A_i, B_i]$를 만들 수 있습니다.
이렇게 하면 간선이 총 $O(N\log N + M)$개 만들어집니다. 사실 앞의 풀이와 별 차이 없습니다.

I. 비행기 타고 가요 - $O((N+M)\log N)$

다른 방법으로 접근하여 시간복잡도를 줄일 수 있습니다. 그래프를 명시적으로 그리지 않고 다익스트라 알고리즘을 사용해 봅시다. $[A_i, B_i]$와 $[C_i, D_i]$를 연결할 때, $[A_i, B_i]$ 쪽에서는 시작 정점인 K와 가장 가까운 단 하나의 정점만이 중요합니다.

I. 비행기 타고 가요 - $O((N+M)\log N)$

알고리즘은 다음과 같습니다.

1. 우선순위 큐에 $(K, K, 0)$을 넣고, 큐가 빌 때까지 반복한다 :
   1. 큐에서 x가 가장 작은 (l, r, x)를 뽑는다.
   2. 구간 $[l, r]$에 남아 있는 정점 v가 없을 때까지 반복한다 :
      1. 정점 K에서 v까지의 최단 경로는 x이다. 답을 저장한다.
      2. $[A_i, B_i]$가 v를 포함하면서 아직 제거되지 않은 모든 i에 대해, 우선순위 큐에 $(C_i, D_i, x + E_i)$를 추가한 뒤 i를 제거한다.
      3. v를 제거한다.

I. 비행기 타고 가요 - $O((N + M)\log N)$

구간에 남아 있는 정점들을 제거하는 과정은 Union-Find나 세그먼트 트리를 사용해 구현할 수 있습니다.

정점을 포함하는 구간들을 제거하는 과정은 세그먼트 트리를 사용해 구현할 수 있습니다.

I. 비행기 타고 가요

- Tag : 최단 경로, 자료 구조
- 출제 및 검수진의 최단 코드 : 1143 bytes
- First to Solve : 뚝심햄구이 (23 min)
- First to Solve (Open) : ainta (78 min)

E. 참 어려운 문제

출제자 : 윤교준

루트 없는 트리의 각 정점에 색이 칠해져 있다. 이 트리에서 루트를 잡았을 때 조상-자식 관계이면서 색깔이 같은 두 정점이 있으면 안 된다. 루트로 가능한 정점들을 모두 구하시오.

E. 참 어려운 문제 - $O(N \log N)$

u와 v를 연결하는 간선이 있다고 합시다. 이 간선을 끊어서 만들어지는 두 서브트리 중 v를 포함하는 쪽에 색깔이 u와 같은 정점이 있다면, u를 포함하는 쪽에 있는 정점들은 루트가 될 수 없다는 것을 알 수 있습니다.

E. 참 어려운 문제 - $O(N \log N)$

모든 간선에 대해 양쪽으로 검사해 보면 됩니다.

어떤 서브트리에 색깔 C인 정점이 있는지는 이분 탐색과 DFS ordering을 사용해 $O(\log N)$에 찾을 수 있습니다.

불가능한 서브트리를 체크하는 것은 $O(1)$에 가능합니다.

E. 참 어려운 문제 - $O(N)$

비슷한 아이디어를 사용해 선형에도 풀 수 있습니다. DFS를 돌면서 각 색깔별로 따로 스택을 관리하는 방법 등이 있습니다. 하지만 훨씬 간단한 선형 풀이가 존재합니다.

E. 참 어려운 문제 - $O(N)$

트리에서 유일한 색깔을 가진 정점의 집합을 S라 합시다. S에 속한 정점들만이 루트가 될 수 있습니다.

뿐만 아니라, 다음 명제가 성립합니다.

Lemma

어떤 $u, v \in S$에 대해, u가 루트가 될 수 있고 u에서 v까지 S에 속한 정점들만을 거쳐 이동할 수 있다면 v 또한 루트가 될 수 있다.

따라서 가능한 정점을 하나만 찾으면 됩니다.

E. 참 어려운 문제 - $O(N)$

정점 1을 루트로 했을 때, 정점 v의 서브트리에 v와 색깔이 같은 정점이 있다면 v를 충돌하는 정점이라 합시다.

충돌하는 정점이 없다면 정점 1은 루트가 될 수 있습니다.

충돌하는 정점이 있을 경우, 루트가 될 수 있는 정점이 있을 수도 있고 없을 수도 있습니다. 일단 있다고 가정합시다.

E. 참 어려운 문제 - $O(N)$

루트가 될 수 있으면서 1번 정점과 가장 가까운 정점을 r이라 합시다. 충돌하는 정점들은 모두 정점 1과 r을 잇는 경로 상에 있어야 합니다.

이때, 충돌하는 정점 중 정점 1과 가장 먼 정점을 x라 하면, x와 r은 이웃해 있게 됩니다! 자세한 증명은 생략합니다.

E. 참 어려운 문제 - $O(N)$

따라서 다음 알고리즘으로 가능한 루트를 찾을 수 있습니다.

1. 충돌하는 정점을 구한다.
2. 충돌하는 정점이 없을 경우, $r = 1$로 놓는다.
3. 있을 경우, 그 중 정점 1과 가장 먼 정점을 구해 x라 하자. 다음 조건을 만족하는 y를 찾아 $r = y$로 놓는다.
   1. y는 x의 자식이다. (이웃해 있다)
   2. y를 루트로 하는 서브트리에 x와 같은 색깔의 정점이 존재한다.
4. r이 루트로 가능한지 판단한다.

E. 참 어려운 문제 - $O(N)$

이와 같이 가능한 루트를 하나 찾았다면, 앞의 명제를 이용해 답을 구할 수 있습니다.

DFS를 2 - 3번만 수행하면 됩니다.

E. 참 어려운 문제

- Tag : 트리, 깊이 우선 탐색
- 출제 및 검수진의 최단 코드 : 816 bytes
- First to Solve : HLD조아 (172 min)
- First to Solve (Open) : cki86201 (29 min)

F. 촛불과 그림자

출제자 : 윤교준

볼록 다각형 안에 또 다른 볼록 다각형이 있고, 촛불이 하나 있다.
안쪽 볼록 다각형에 의해 만들어지는 그림자의 넓이를 구하시오.

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

구현해야 할 것들은 다음과 같습니다.

1. 다각형의 내부 / 외부 판별하기
2. 다각형 외부의 점에서 다각형에 그은 접선 구하기
3. 다각형과 반직선의 교점 구하기
4. 넓이 구하기

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

1. 다각형의 내부 / 외부 판별하기

   다각형의 아래쪽과 위쪽 껍질을 나누면 구현이 편해집니다(?)

   이분 탐색을 사용해 $O(\log N)$에 할 수 있습니다.

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

2 다각형 외부의 점에서 다각형에 그은 접선 구하기

점을 P, 다각형의 꼭짓점들을 $V_1, V_2, \cdots, V_N$이라 합시다. 다각형과 $\overleftrightarrow{PV_1}$의 두 교점 중 V_1이 아닌 점이 V_k와 V_{k+1} 사이에 있다면, 접점은 V_1과 V_k 사이에 하나, V_{k+1}과 V_N 사이에 하나 있게 됩니다. 이제 열심히 이분 탐색을 합시다.

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

3 다각형과 반직선의 교점 구하기

반직선을 $\overrightarrow{PX}$라 합시다. k가 주어졌을 때 두 반직선 $\overrightarrow{PV_1}$과 $\overrightarrow{PV_k}$가 만드는 영역에 점 X가 포함되는지 구할 수 있고, 이를 이용해 이분 탐색을 하면 교점이 위치하는 변을 구할 수 있습니다. 연립방정식을 풀어 교점을 구하면 됩니다.

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

4 넓이 구하기

$\overrightarrow{OV_i} \times \overrightarrow{OV_{i+1}}$의 prefix sum을 미리 저장해 놓으면 넓이를 $O(1)$에 구할 수 있습니다.

F. 촛불과 그림자 - $O(N + M + Q(\log N + \log M))$

어때요, 참 쉽죠?

F. 촛불과 그림자

- Tag : 기하
- 출제 및 검수진의 최단 코드 : 2615 bytes
- First to Solve : ??? (? min)
- First to Solve (Open) : ??? (? min)

평가

평균 C++ 코드 길이(F 제외) : 700 bytes

평균 C++ 코드 길이 : 847 bytes

오! 클린한 대회!

평가

문제가 어려웠는데, 다들 굉장히 잘 풀어주셨습니다.

C가 예상보다 훨씬 많이 풀림 ㄷㄷ

G는 예상보다 너무 적게 풀림 ㅠㅠ

난이도 역순으로 문제 푸는 거 잘 봤습니다 ㅎㅎㅎ

경곽의 미래는 밝군요!

특별상 시상

특별상 1

선정 방식 : 다른 특별상 수상팀 제외, 완전 랜덤하게 선정

상품 : 인형 + 잡다한거

특별상 1

선정 방식 : 다른 특별상 수상팀 제외, 완전 랜덤하게 선정

상품 : 인형 + 잡다한거

??? 팀

???, ???, ???

축하합니다!

특별상 2

선정 방식 : 한 문제를 가장 많이 시도한 팀

상품 : 인형 + 잡다한거

특별상 2

선정 방식 : 한 문제를 가장 많이 시도한 팀

상품 : 인형 + 잡다한거

"서쿠니" 팀

18072 양석훈, 18051 박지환

축하합니다!

특별상 3 - NEXON 특별상

선정 방식 : ”D. 다오와 데이트” 문제를 가장 먼저 푼 팀

상품 : 인형 + 잡다한거

특별상 3 - NEXON 특별상

선정 방식 : "D. 다오와 데이트" 문제를 가장 먼저 푼 팀

상품 : 인형 + 잡다한거

"1111" 팀

18050 박준호

축하합니다!

순위 공개

3위 공개

상품 : 특별상 상품 + 마이크로닉스 마이크 헤드셋

3위 공개

상품 : 특별상 상품 + 마이크로닉스 마이크 헤드셋

"의문부호의 꼭두각시" 팀

18021 김석표, 18080 이민규, 18081 이민준

6 문제 / 703 min

축하합니다!

2위 공개

상품 : 특별상 상품 + 제닉스 타이탄 게이밍 마우스 + 제닉스 타이탄 청축 LED 게이밍 기계식 키보드

2위 공개

상품 : 특별상 상품 + 제닉스 타이탄 게이밍 마우스 + 제닉스 타이탄 청축 LED 게이밍 기계식 키보드

"KR815KR" 팀

17003 고동현, 17074 이강민, 17082 이민제

7 문제 / 960 min

축하합니다!

1위 공개

상품 : 특별상 상품 + 삼성 외장SSD 500GB

1위 공개

상품 : 특별상 상품 + 삼성 외장SSD 500GB

"뚝심햄구이" 팀

18089 이종영, 18070 안지민, 18115 최은수

11 문제 / 2130 min

축하합니다!