20.09.23 동아리 풀이

선린인터넷고등학교 소프트웨어과

30610 나정휘

https://JusticeHui.github.io

JusticeHui

문제 목록

- A. 주때의 자소서 쓰기
- B. 편지 꼭 해다오 - 제1회 논산 코드 페스티벌
- C. 박 터뜨리기 - 2020 KOI 1차
- D. 햄버거 분배 - 2020 KOI 1차
- E. 사무실 이전 - 2018 서울대 프로그래밍 대회
- F. Building Bridges - 2017 CEOI(중앙 유럽 정보 올림피아드)
- G. 최단경로와 쿼리 - Good Bye, BOJ 2019
- H. 케이크 3 - 2019 JOISC(일본 국가대표 선발고사)

JusticeHui

문제 난이도

- A. 주때의 자소서 쓰기 - 중간(사전지식: MCMF)
- B. 편지 꼭 해다오 - 매우 쉬움
- C. 박 터뜨리기 - 매우 쉬움
- D. 햄버거 분배 - 매우 쉬움
- E. 사무실 이전 - 중간(사전지식: 센트로이드, 이진 트리 변환)
- F. Building Bridges - 쉬움(사전지식: CHT)
- G. 최단경로와 쿼리 - 어려움
- H. 케이크 3 - 어려움

JusticeHui

문제 태그

- A. 주때의 자소서 쓰기 - MCMF
- B. 편지 꼭 해다오 - 완전 탐색
- C. 박 터뜨리기 - 수학
- D. 햄버거 분배 - 이분 매칭 or 그리디
- E. 사무실 이전 - 센트로이드, 이진 트리 변환 테크닉
- F. Building Bridges - 컨벡스헐 트릭
- G. 최단경로와 쿼리 - 다익스트라, 분할정복
- H. 케이크 3 - 그리디, DnC Opt, PST

JusticeHui

A. 주때의 자소서 쓰기 (1)

- “각 문항에 한 개 이상의 스토리가 들어가야 한다” 라는 조건을 빼고 생각해보자.

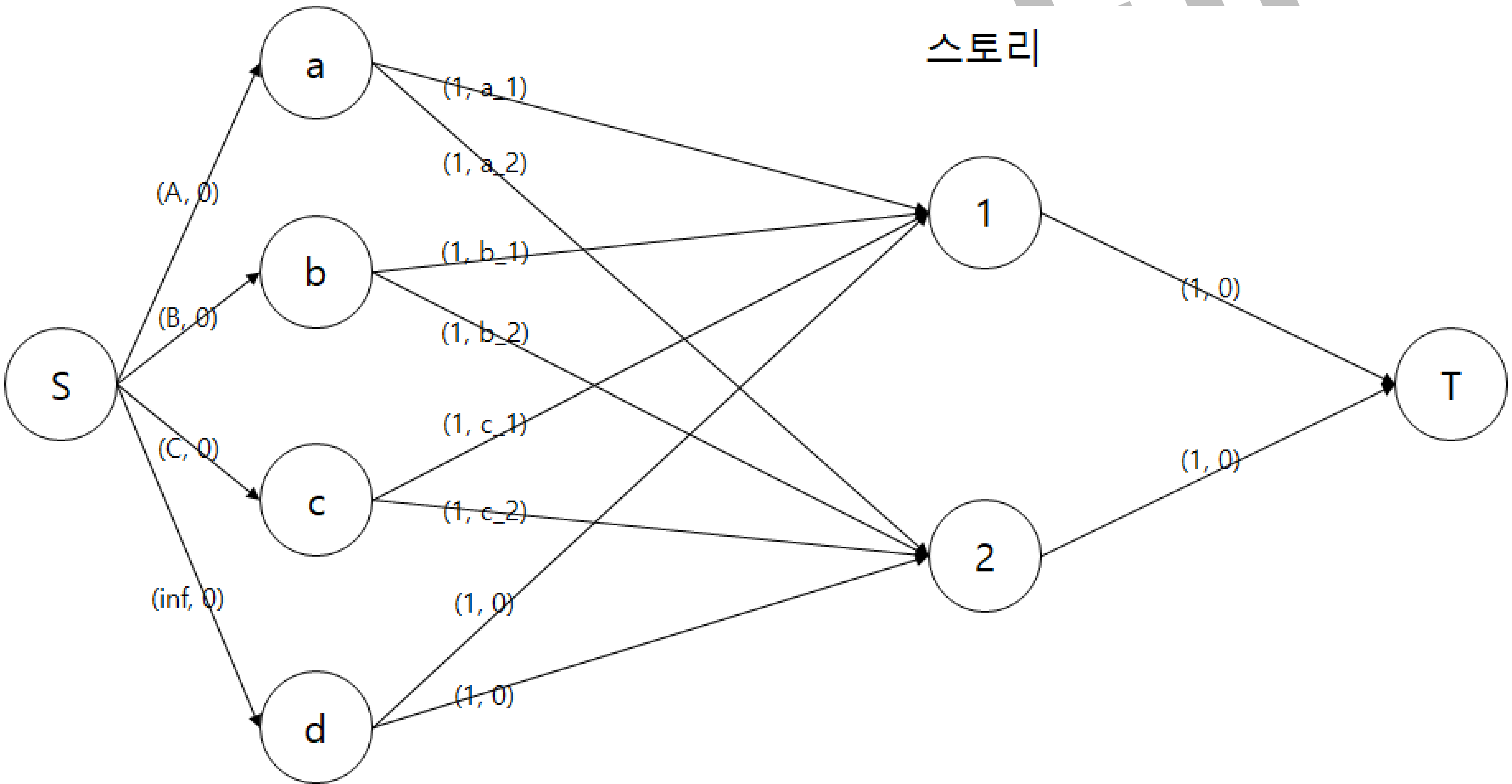
- 1번 문항에 a개, 2번 문항에 b개, 3번 문항에 c개가 들어가고
- 나머지 D(= N-a-b-c)개는 버린다. ($a \le A$, $b \le B$, $c \le C$)

- MCMF로 모델링할 수 있다.

JusticeHui

A. 주때의 자소서 쓰기 (2)

- (c, d)는 용량이 c이고 비용이 d인 간선을 의미한다.



A. 주때의 자소서 쓰기 (3)

- “각 문항에 한 개 이상의 스토리가 들어가야 한다”

- N개의 문항 중 3개를 따로 빼서
- 미리 각 문항에 하나씩 넣어야 한다.

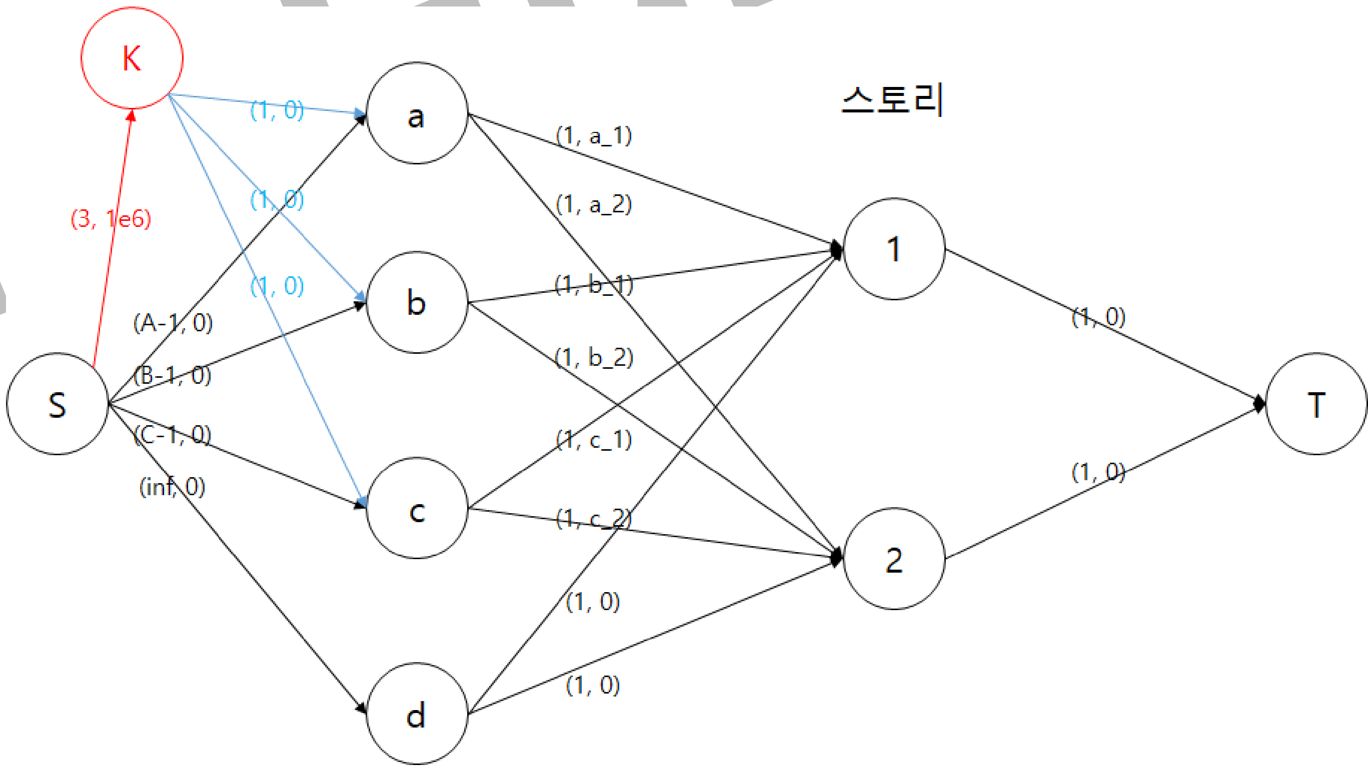
- 잘 생각해보면 이것도 MCMF로 모델링할 수 있다.

JusticeHui

A. 주때의 자소서 쓰기 (4)

- 가중치를 "충분히 크게" 만들어주면 무조건 그 간선을 먼저 사용하게 된다.

- 정답에 3e6이 더해졌으므로
- 3e6을 빼서 출력하면 된다.



Justi

A. 주때의 자소서 쓰기 (5)

- DP로 풀 수도 있다는데 저는 어떻게 하는지 모릅니다.

- Aliens Trick으로도 풀 수 있다는데 이건 뇌절인듯

JusticeHui

B. 편지 꼭 해다오 (1)

- 로컬에서 완전탐색을 돌리면
- 5글자 짜리 해를 찾을 수 있다.

- 이걸 왜 못 풀지

JusticeHui

C. 박 터뜨리기 (1)

- $N' = N - K(K+1)/2$로 정의하자.
  - Case 1) $N' < 0$이면 정답은 -1이다.
  - Case 2) $K \mid N'$이면 정답은 $K-1$이다.
  - Case 3) 1, 2에 해당하지 않으면 정답은 K이다.

JusticeHui

D. 햄버거 분배 (1)

- Solution 1.
  - 이분 매칭
  - chk배열을 매번 초기화하면 TLE
  - chk배열 초기화 안 하고 timeframe을 잘 관리해주면 AC

JusticeHui

D. 햄버거 분배 (2)

- Solution 2.
  - 왼쪽에 있는 사람부터 보면서
  - 먹을 수 있는 가장 왼쪽 햄버거를 먹으면 된다.

JusticeHui

E. 사무실 이전 (1)

- 모든 경로를 고려한다 -> Centroid Decomposition

- 센트로이드 C를 지나는 모든 경로를 처리하고
- 나머지는 분할정복 과정에서 처리하자.

JusticeHui

E. 사무실 이전 (2)

- 두 정점 u v를 잇는 경로가 c를 지난다 =
- 루트가 c인 트리에서, u와 v가 서로 다른 서브 트리에 속한다.

- 루트가 c인 트리에서 u v의 깊이가 d_u d_v라면
- 경로 길이의 제곱은 $d_u^2 + d_v^2 + 2 d_u d_v$

JusticeHui

E. 사무실 이전 (3)

- 서브 트리 T1에 있는 정점 u1, u2, … , uk의 깊이가 각각
  - d1, d2, … , dk
- 서브 트리 T2에 있는 정점 v1, v2, … , vs의 깊이가 각각
  - e1, e2, … , es

- ui에서 vj로 가는 모든 경로 길이 제곱의 합은
- s×sum(di^2) + t×sum(ej^2) + 2sum(di)sum(ej)

JusticeHui

E. 사무실 이전 (4)

- 각 서브 트리마다
  - 직원 수, 직원 깊이 합, 직원 깊이 제곱 합
  - 후보 역 수, 후보 역 깊이 합, 후보 역 깊이 제곱 합
- 을 알고 있으면 답을 구할 수 있음.

- 센트로이드 c를 루트로 한 트리에서
- 서브 트리가 T1, T2, … , Tk라고 하면 O(k^2 log N)이다.
- k는 최대 N-1이라서 TLE

JusticeHui

E. 사무실 이전 (5)

- k를 최대 3으로 만들 수 있다.

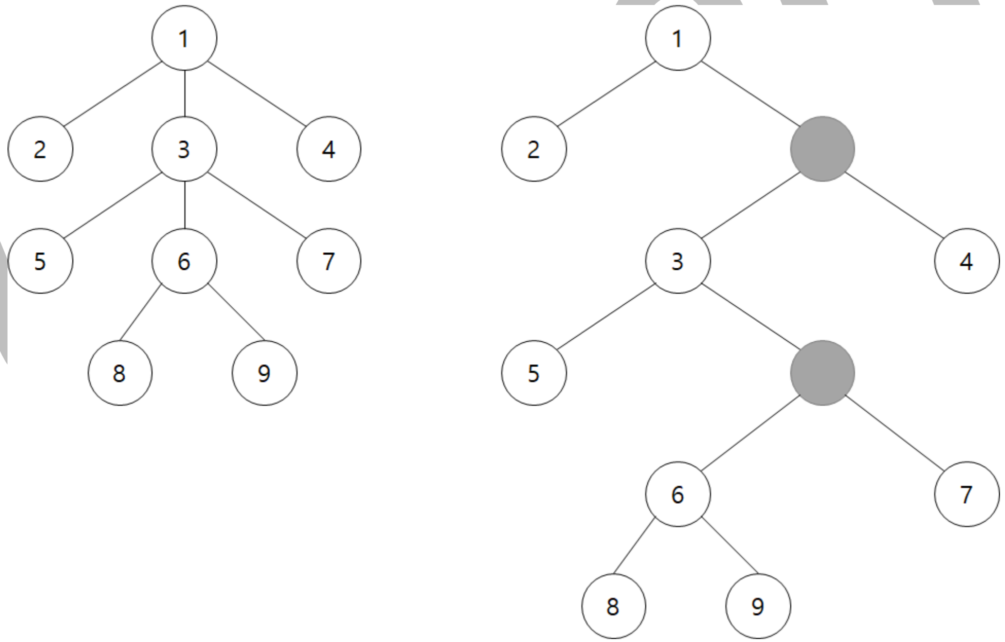
- 트리에 N개의 더미 정점을 추가해서
- 정점 간의 거리 관계를 유지하며
- 이진 트리로 바꿔주는 테크닉이 있다.

- 이진 트리의 차수는 최대 3이므로 O(N log N)에 문제를 풀 수 있다.
- 앞에 상수가 36정도 붙는 것 같은데, 할만하다.

JusticeHui

E. 사무실 이전 (6)

- 이진 트리로 바꾸는 건 이런 식으로 하면 된다.
- 더미 노드로 내려가는 간선의 가중치를 0으로 설정하면 된다.



F. Building Bridges (1)

- 점화식을 세워보자.
- min함수 내부에 있는 Hi를 x로 치환하면 min함수 내부가 ax+b꼴이 된다.
- 일차함수가 여러 개 있을 때 최솟값을 구하는 것은 Convex Hull Trick

$$D_i = \min_{1 \le j < i} (S_{i-1} - S_j + (H_i - H_j)^2 + D_j)$$

$$D_i = \min_{1 \le j < i} (S_{i-1} - S_j + H_i^2 - 2H_iH_j + H_j^2 + D_j)$$

$$D_i = \min_{1 \le j < i} (-2H_jH_i - S_j + H_j^2 + D_j) + S_{i-1} + H_i^2$$

Hui

G. 최단 경로와 쿼리 (1)

- M에 비해 N이 너무 작은 점을 이용해서 문제를 풀자.

- L번째 열~R번째 열 안에 있는 쿼리를 해결한다고 하자.
  - L과 R의 중간 지점 M(= (L+R)/2)를 잡자.
  - 만약 쿼리의 시작점과 끝점이 M열을 기준으로 서로 반대쪽에 있다면
  - **최단 경로는 항상 M번째 열에서 최소 한 칸을 지난다.**

  - M열에는 N(≤ 5)개의 칸이 있으므로 Dijkstra N번으로 M열을 지나는 모든 경로의 최단 거리를 구할 수 있다.

JusticeHui

G. 최단 경로와 쿼리 (2)

- 이제, M열을 지나지 않는 쿼리를 처리해야 한다.
  - 최단 경로가 M열을 지나지 않는다는 것은
  - 쿼리의 시작 점과 끝 점이 한 쪽에 있다는 것을 의미한다.
  - [L, M-1] [M+1, R] 구간에 대해 각각 분할 정복을 해주면 된다.

  - O(NM log (NM))짜리 다익스트라를 N번 돌리는 과정을
  - O(log M)번 반복한다.

- O(NM log (NM) * N * log M)
- = O(N^2M log(NM)log(M))에 문제를 풀 수 있다 ^^7

JusticeHui

H. 케이크 3 (1)

- 일단 어떻게 M개를 잘 선택했다고 하자.
  - Vi의 합은 상수
  - M개를 잘 배치해서 인접한 Ci의 차이를 최소화하면 된다.
  - 잘 생각해보면 Ci를 오름차순으로 정렬하는 것이 최적임을 알 수 있고
    - 이때 인접한 원소 차이의 합은 2*(MAX - min)이다.
    - Ci를 2배씩 해주는 것이 정신 건강에 이롭다.

- 이제 문제는
  - M개를 잘 선택해서
  - (V의 합 - C의 최대 + C의 최소)를 최대화하는 문제로 바뀌게 된다.

JusticeHui

H. 케이크 3 (2) - Subtask 1

- O(N^3) 풀이를 먼저 알아보자.

- 선택할 Ci의 최대/최소를 고정시키고
- 선택 가능한 원소 중 V가 가장 큰 M개를 선택하면 된다.

- 구현 방식에 따라 O(N^3) ~ O(N^3 log N)정도 걸린다.
- 실제 대회에서는 5점을 받을 수 있었다.

JusticeHui

H. 케이크 3 (3) - Subtask 2

- O(N^2 log N) 풀이를 알아보자.

- 각 조각을 pair(Ci, Vi)순으로 정렬하자.
- O(N^3) 풀이에서는 Ci의 최대/최소를 고정했는데
  - (Ci, Vi)순으로 정렬했으니 i < j인 i와 j를 선택하는 것과 똑같다.

- 시작점 i를 정한 뒤, j를 하나씩 증가시키면서
- multiset이나 우선순위 큐로 Vi가 가장 큰 M개를 관리해주면 된다.

- 실제 대회에서는 24점을 받을 수 있었다.

JusticeHui

H. 케이크 3 (4) - Subtask 3

- 100점 짜리 풀이를 알아보자.

- 앞에서 소개한 두 풀이처럼 O(N^2)개의 구간을 모두 보는 것으로는 O(N^2)보다 빠르게 풀 수 없다.
  - 고려하는 구간의 개수를 줄여야 한다.

JusticeHui

H. 케이크 3 (5) - Subtask 3

- Subtask 2에서 사용한 풀이를 다시 보자.

- 시작점 i를 고정했을 때
- 정답을 찾을 수 있는 가장 작은 j값을 opt[i]라고 정의하자.

- $opt[i] \le opt[i+1]$이 성립한다는 것은 직관적으로 알 수 있다.
- 최적 위치가 단조증가하므로 DnC Opt를 적용할 수 있다.

JusticeHui

H. 케이크 3 (6) - Subtask 3

- 원소들이 C_i 기준으로 정렬되어 있으므로
  - $i \in [s, e]$인 i에 대해, V_i들 중 가장 큰 M개의 합 구하기
  - 이 연산만 빠르게 할 수 있으면 된다.

- 이 연산은 Persistent Segment Tree를 이용해 $O(\log N)$에 할 수 있다.

- $T(N) = 2T(N/2) + O(N \log N)$이므로 $O(N \log^2 N)$에 문제를 풀 수 있다.

JusticeHui