

Day2 그래프 이론

선린인터넷고등학교 소프트웨어과

30610 나정휘

문제 목록

- BOJ2251 물통
- BOJ1516 게임 개발
- BOJ1753 최단경로
- BOJ16188 달빛 여우
- BOJ1854 K번째 최단경로
- BOJ2848 알고스팟어
- BOJ16297 Eating
- BOJ2595 배수
- BOJ1848 동굴 탐험
- BOJ7907 Road Race

물통

- 단순 BFS 문제
- 각 물통에 있는 물의 용량을 정점으로 잡고 열심히 구현

게임 개발

- DAG의 최장 경로
- DP로 대충 구하면 됨

최단경로

- 다익스트라 구현 문제

달빛 여우

- 여우는 단순 다익스트라로 구할 수 있음
- 늑대는 각 정점을 2개로 분할한 다음 다익스트라를 돌리면 편함

K번째 최단경로 찾기

- 기존 다익스트라 : $dst[v]$ 배열
- K번째 최단거리를 구하는 다익스트라 : $dst[v]$ 힙(??)
- min heap인 $dst[v]$ 의 크기가 K가 되도록 유지하면 됨
 - $dst[v].size() \leq k$ 인 경우에는 삽입
 - $dst[v].size() == k$ 인 경우에는 삽입하고 최댓값 제거

알고스팟어

- 각 단어를 앞에서부터 비교하면서 처음으로 달라지는 위치를 찾자.
- 그 위치에 있는 글자들의 순서를 알 수 있다.
 - $g[s][e] = 1$: s보다 e가 뒤에 나옴
 - Floyd 돌려주면 모든 글자쌍의 순서를 알 수 있음
- 잘 정렬하자.

Eating Everything Efficiently

- DAG에 대한 DP
- $D(v)$ = v 까지 갔을 때 최댓값

배수

- 사용할 숫자들을 고정하자. -> 총 $2^{10} = 1024$ 개의 경우의 수
- 정점이 N개인 그래프를 만들자.
 - i번 정점은 (현재 수) % N = i를 의미한다.
 - 적당한 위치에서 시작해서 0에 도착하면 된다.

배수

- 현재 수 T 에 i 를 붙인다
 - $T * 10 + i$
 - $(T \% N)$ 번 정점에서 $((T * 10 + i) \% N)$ 번 정점으로 이동
- 사용할 숫자들을 고정시키고 그래프를 잘 만든 뒤
- BFS로 최단 경로를 잘 찾으면 된다!

동굴 탐험

- 1로 다시 돌아오기 직전에 방문하는 정점 i 를 고정하자.
 - 1과 i 를 잇는 간선을 제거하면 (1- i 최단거리) + (1- i 간선 가중치)
 - 다익스트라 N 번 돌려서 $O(NM \log N)$

동굴 탐험

- 1로 다시 돌아오기 직전에 방문하는 정점 i 를 고정하자.
 - 1과 i 를 잇는 간선을 제거하면 $(1-i \text{ 최단거리}) + (1-i \text{ 간선 가중치})$
 - 다익스트라 N 번 돌려서 $O(NM \log N)$
- 1에서 시작해 처음으로 방문하는 정점 j 를 고정하자.
 - $1-j$ 간선 제거 $(1-j \text{ 간선 가중치}) + (j-i \text{ 최단거리}) + (1-i \text{ 간선 가중치})$
 - 서로 다른 모든 (i, j) 쌍에 대해 구하면 됨

동굴 탐험

- 임의의 두 정수 i, j 가 다르다 \rightarrow 최소 비트 하나가 다르다
- 1번과 연결된 정점을 V_1, V_2, \dots, V_k 라고 하자.
 - i 의 x 번째 비트가 켜져있으면 V_i 를 A그룹, 꺼져있으면 B그룹에 넣고
 - 1-A그룹 간선 제거하고 A그룹-B그룹 최단 거리 구해주면 됨
 - 모든 비트에 대해 다 하면 $O(M \log N \log N)$

Bytean Road Race

- 두 점을 모두 지나지 않는다 = 한 점을 지나면 다른 한 점을 못 지난다.
- DAG이므로 $A \rightarrow B$ 경로가 있으면 $B \rightarrow A$ 경로는 없다.

Bytean Road Race

- 두 점을 모두 지나지 않는다 = 한 점을 지나면 다른 한 점을 못 지난다.
- DAG이므로 $A \rightarrow B$ 경로가 있으면 $B \rightarrow A$ 경로는 없다.
- DAG에서는 위상 정렬을 할 수 있다.
 - 그래프의 형태, 탐색 순서에 따라 위상 정렬의 결과가 다양할 수 있다.
- U, V를 모두 지나고 U 다음에 V를 지나는 경로가 존재한다고 하면
 - 어떻게 위상 정렬을 해도 항상 U가 V보다 먼저 나온다.
- 모두 지나는 경로가 존재하지 않는다면
 - 탐색 순서에 따라 U가 먼저 나올 수도, V가 먼저 나올 수도 있다.

Bytean Road Race

- 최대한 다른 결과가 나오도록 위상 정렬을 한 뒤
- 두 정점의 전후 관계를 살피면 됨

Bytean Road Race

- 최대한 다른 결과가 나오도록 위상 정렬을 한 뒤
- 두 정점의 전후 관계를 살피면 됨
- 아래로 가는 간선 / 오른쪽으로 가는 간선
 - Method 1 : 아래로 가는 간선을 먼저 처리
 - Method 2 : 오른쪽으로 가는 간선 먼저 처리
- 위상 정렬 : $O(N+M)$, 쿼리 $O(1)$, 총 $O(N+M+K)$