

인터넷 설치.

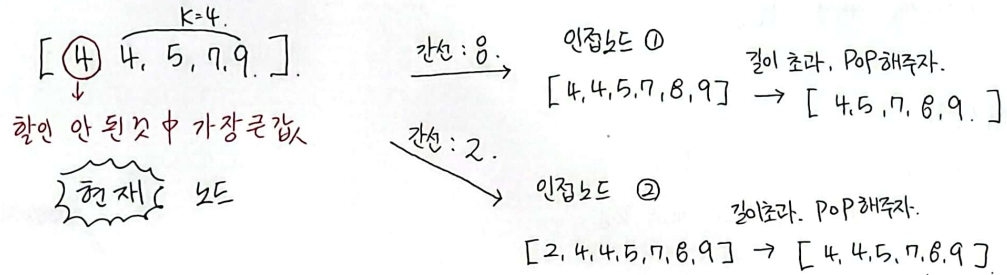
1 ~ N 까지 가는 경로에서 가장 비싼 K개를 할인 (0원으로)

할인이 안된 비몽 중 가장 큰 것의 \min 값?

내 풀이 (메모리 초과).

1. Node 클래스에 원소의 큐를 필드로 추가. (추가와 삭제가 빠르니까)

2. 우선 순위 큐의 크기를 $k+1$ 개로 유지하면서 인접노드로 이동.



3. 백트래킹 (일종의 visited)

Node 744. to

다익스트라 우선순위 큐에서 방문 받은 Node 의 $pf.peek()$

Node 744. from

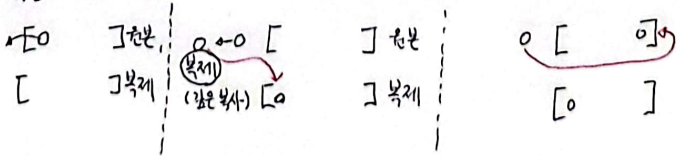
값이 $dptable(answer)$ 의 값보다 크면 바꿔주기.
(= continue)

Answer

[부록]

우선 순위 큐의 복제.

• 기존의 큐는 사이즈를 재고, 뽑아서 복제하고, 다시 넣어서 주면 됐는데.



○ 우선순위 큐는 뺏고 넣으면 무조건 다시 맨앞으로 들어간다. j.

배열을 새로 생성해서 전부 뺀고 다시 다 넣어줘야 함

→ 이것때문에
메모리 크라임 등

정답 폭이.

- ? 이분탐색.
- 정렬되어 있어야 함.
 - 최소 / 최대를 알아야 함.

무엇을

이분탐색 할 것인가?

- 간선?
- 노드?
- 거리?
- 최대 간선?

⇒

↓ mid 값이 됨.
"정답"을 이분탐색하자.
↳ K개 확인 후에 가장 큰 값.

★ 예외

특정값에서 Pass/nonPass 를 알았을 때. 특정값보다 큰/작은 값들이 모두 탈락된다면
이분탐색 가능한 부분.

다익스트라

(left, mid, right.)

↓
이 값으로 테스트 해볼 것임.

조건문처리.

mid 값보다 큰 간선은 가중치를 1, 작은 간선은 가중치를 0 으로 보고,
최단거리를 구하면?

dist[] 에는 mid 값보다 컸던 간선들의 "개수"가 저장됨.

아예 K (확인 개수) 보다 크면 nonPass (left = mid + 1)

K (") 보다 작거나 같으면 Pass (right = mid)