

week 5

2024-08-01

타임머신

분명히 맞는데 자꾸 틀려서 한참을 들여다보다가

거리를 담는 배열을 long long 으로 선언해야 한다는걸 알게됨

타임머신

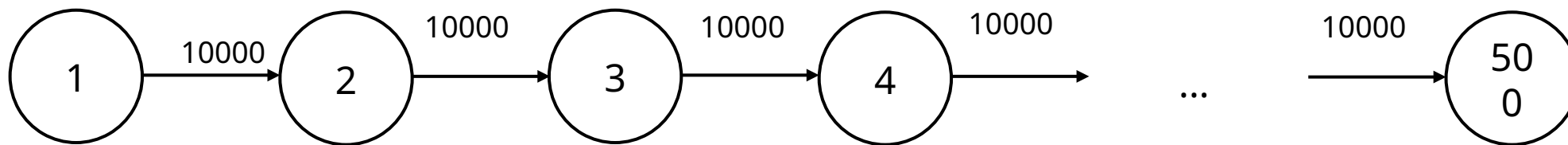
문제에서 ,

도시의 개수는 최대 500 개 , 간선의 개수는 최대 6000 개 ,
간선의 가중치는 $[-10000, 10000]$ 이다

최악의 경우를 생각해 본다면

타임머신

$500 * 10000 \Rightarrow 5,000,000$ 일 거라고 생각했음



$5,000,000$ 은 int 의 최댓값에 한참 못 미치는 값

근데 음의 가중치가 있어서 다시 생각해봐야 함

타임머신

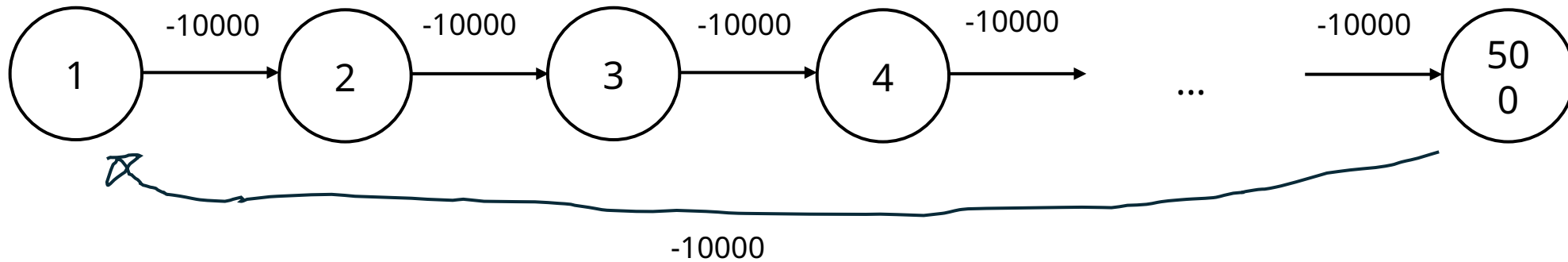
```
if(dist[j] != INF && dist[next] > dist[j] + d) {  
    dist[next] = dist[j] + d;  
}
```

코드에서

next 까지의 거리보다 , 현재까지의 거리 + 가중치가 작다면 갱신해줌

간선의 가중치가 전부 -10000 라면 계속 갱신됨

타임머신



그래프가 위와같이 그려질 수도 있다 .

그래서 거리의 최소값은 $500 * 500 * -10000$, 최대값은 $500 * 10000$ 의 범위를 가짐

$[-2,500,000,000 \quad 5,000,000]$

타임머신

`[-2,500,000,000 5,000,000]`

-2,500,000,000 은 int 의 최솟값을 넘어선다

int 는 대략 $\pm 2.1 \times 10^9$ 정도라고 보면 됨 (2,147,483,647)

문제에서 가끔 1,000,000,007 로 나눈 나머지를 구하라고 하는데

1,000,000,007 은 두배해도 간당하게 int 를 넘지 않음

long long 은 대략 $\pm 9 \times 10^{18}$ 정도

그래서 longlong 의 최대값으로 보통 $1e18$ 을 사용함 그 이상 넘어갈 일은 별로 없음

박스나누기

dp[a][b] 는 dp[b][a] 와 정확히 똑같은 상태임 .

그래서 for 문을 사용할 때 [1.. m - 1] 을 하지
않고

절반까지만 돌면서 작은 값을 왼쪽에 주면

조금 더 효율적으로 해결할 수 있다

```
int go(int n, int m) {  
    if(n == 1 && m == 1) return 1;  
  
    int &ret = dp[n][m];  
    if(ret != -1) return ret;  
    ret = 1;  
    for(int i = 1; i <= m / 2; i++) {  
        ret = min(ret , 1 - go(i, m - i));  
    }  
    for(int i = 1; i <= n / 2; i++) {  
        ret = min(ret , 1 - go(i, n - i));  
    }  
    return ret;  
}
```