



서울에서 경산까지

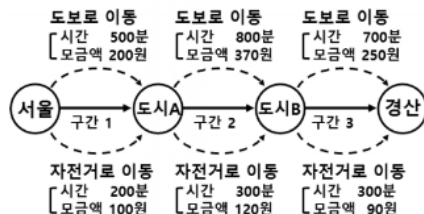
소프트웨어학과 이지우

BOJ 14863

문제

배우 한정을 씨는 이번 여름에 서울에서 경산까지 자선 여행을 하면서 모금 활동을 진행할 계획이다. 자선 여행에서 거쳐 가게 될 도시의 개수와 순서는 미리 정해져 있으며, 자선 여행은 서울에서 시작하여 각 도시를 정해진 순서대로 단 한 번씩 방문한 후 경산에서 끝난다. 서울을 제외한 도시의 개수를 N 이라 하자. 이때 서울에서 두 번째 도시까지 가는 구간을 구간 1, 두 번째 도시부터 세 번째 도시까지 가는 구간을 구간 2와 같이 부르기로 하며, 마지막 목적지인 경산에 도착하는 구간을 구간 N 이라 하자. 즉, 구간의 전체 개수는 N 이다. 구간 사이의 이동은 도보 혹은 자전거 어느 한 쪽을 이용하게 되는데, 각 구간에는 도보로 이동할 때 걸리는 시간(분), 이때 얻게 되는 모금액(원), 자전거로 이동할 때 걸리는 시간(분), 이때 얻게 되는 모금액(원)이 정해져 있다.

예를 들어, 서울과 경산 사이에 2개의 도시가 있는 다음과 같은 경우($N = 3$)를 생각해 보자.



인접한 도시 사이를 **도보로 이동하는지 자전거로 이동하는지**에 따라 전체 모금액이나 걸리는 시간에 차이가 생기게 된다. 한정을 씨는 **전체 모금액을 가능한 많이 얻는 방법**을 찾고 싶어 한다. 위의 예에서는 시간이 충분하다면 모든 구간을 도보로 이동하는 것이 모금액을 최대로 하는 방법이며, 모금액은 $200+370+250 = 820$ 원, 여행에 걸리는 시간은 $500+800+700 = 2,000$ 분이다.

그러나 한정을 씨는 바쁜 스케줄로 인해 자선 여행을 위해 보낼 수 있는 시간이 K 분(K 는 자연수)으로 한정되어 있다. 위의 예에서 만약 $K = 1,650$ 이라면, 1, 2번 구간은 도보로 이동하고 3번 구간은 자전거로 이동하여 모금액을 660원으로 하는 것이 가장 좋은 방법이며, 이때 걸리는 시간은 1,600분이다.

위와 같이 각 구간별로 도보 및 자전거로 이동하는 경우 걸리는 시간과 모금액이 주어질 때, **제한시간 이내로 서울에서 경산까지 여행하면서 모금할 수 있는 최대 금액**을 찾는 프로그램을 작성하시오. (제한시간 이내에 여행하는 방법은 항상 존재한다.)

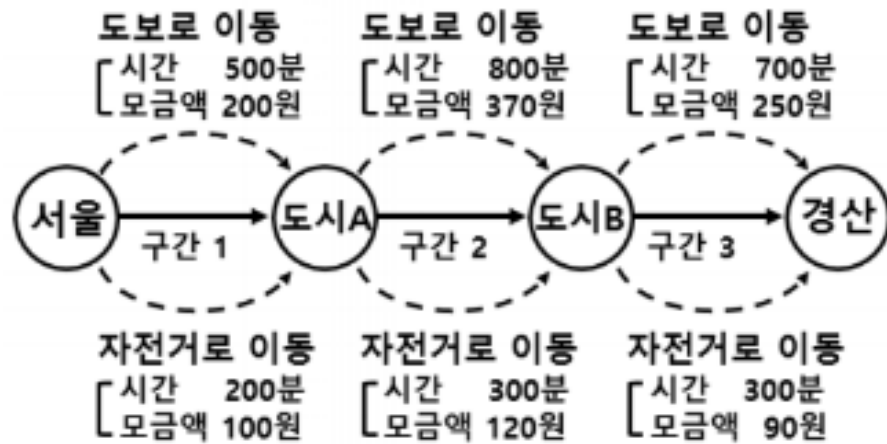
접근

입력

표준 입력으로 다음 정보가 주어진다. 첫 번째 줄에는 두 자연수 N 과 K 가 공백으로 분리되어 주어진다($3 \leq N \leq 100$, $0 < K \leq 100,000$). 두 번째 줄에는 구간 1을 도보로 이동할 때 걸리는 시간(분), 이때 얻게 되는 모금액(원), 자전거로 이동할 때 걸리는 시간(분), 이때 얻게 되는 모금액(원)을 나타내는 네 개의 자연수가 차례로 공백으로 분리되어 주어진다. 세 번째 줄부터 $N+1$ 번째 줄도 마찬가지로 각 줄마다 네 개의 자연수가 주어지며, 입력은 총 $N+1$ 줄로 구성된다. 두 번째 줄부터 $N+1$ 번째 줄에 주어지는 숫자들 중 시간을 나타내는 숫자(각 줄의 첫 번째, 세 번째 숫자)는 10,000 이하의 자연수, 모금액을 나타내는 숫자(각 줄의 두 번째, 네 번째 숫자)는 1,000,000 이하의 자연수들이다.

출력

표준 출력으로 K 분 이내로 여행하면서 모금할 수 있는 최대 금액을 출력한다. (K 분 이내에 여행하는 방법은 항상 존재한다.)



DP문제

현재 도시에서 최대의 모금액을 만들기 위해서는
직전 도시에서의 최대 모금액을 알아야 한다



T시에 도시B에 위치해 있다면,

- *도시A에서 T-800시에 도보로 출발했거나,
- *도시A에서 T-300시에 자전거로 출발한 것.

$$DP[\text{도시B}][T] = \text{MAX}(DP[\text{도시A}][T-800] + 370\text{원}, DP[\text{도시A}][T-300] + 120\text{원})$$

3 1650

500 200 200 100

800 370 300 120

700 250 300 90

byWalkTo[city]: { 도보로 갈 때 걸리는 시간, 모금액 }
byBikeTo[city]: { 자전거로 갈 때 걸리는 시간, 모금액 }



```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);

int n, k;
cin >> n >> k;

vector<pii> byWalkTo(n+1);
vector<pii> byBikeTo(n+1);

for(int i=1; i<=n; i++) {
    int wt, wm;
    int bt, bm;

    cin >> wt >> wm >> bt >> bm;

    byWalkTo[i] = {wt, wm};
    byBikeTo[i] = {bt, bm};
}
```

```

vector<vector<int>> dp(n+1, vector<int>(k+1));
for(int city=1; city<=n; city++) {

    int wt = byWalkTo[city].first;
    int wm = byWalkTo[city].second;

    int bt = byBikeTo[city].first;
    int bm = byBikeTo[city].second;

    for(int time=0; time<=k; time++) {

        dp[city][time] = INT_MIN;
        if(time-wt >= 0) {
            dp[city][time] = max(dp[city][time], dp[city-1][time-wt] + wm);
        }

        if(time-bt >= 0) {
            dp[city][time] = max(dp[city][time], dp[city-1][time-bt] + bm);
        }
    }
}

cout << dp[n][k] << endl;

```

City에 time시간에 도착했을 때의 최대 모금액.

특정 시간에는 city 도시에 도착하지 못하는 경우가 있기 때문에 아주 작은 음수로 초기화

도보로 왔을 때의 모금액 더하기

자전거로 왔을 때의 모금액 더하기

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이
49159863	january	 14863	100점	42004 KB	56 ms	C++17 / 수정	1168 B
49159113	january	 14863	틀렸습니다			C++17 / 수정	977 B
49159027	january	 14863	틀렸습니다			C++17 / 수정	976 B
49145110	january	 14863	틀렸습니다			C++17 / 수정	1035 B