



편의점 맥주 4캔 만원

ALPS 2020년 7월 내부대회 D번

출제자: 고주연

풀이

맥주 가격이 최소 2500원 이상이므로, 4캔 만원 행사를 적용하지 않은 채 날개로 (제 가격으로) 맥주를 구매하는 가능한 경우는 각각 맥주 0개, 1개, 2개, 3개 구입 - 총 4가지입니다. 따라서 날개로 구매할 맥주 수 4가지 case 모두에 대해 행복도를 다음과 같이 계산해주면 됩니다.

- 1) 맥주를 0개 고를 수 있는 서로 다른 모든 경우에 대해 날개로 0개 고른 뒤,
남은 돈과 남은 맥주들로 4캔 만원 행사를 적용하여 맥주를 구매하여 얻을 수 있는 최대 행복도
- 2) 맥주를 1개 고를 수 있는 서로 다른 모든 경우에 대해 날개로 1개 고른 뒤,
남은 돈과 남은 맥주들로 4캔 만원 행사를 적용하여 맥주를 구매할 때 얻을 수 있는 최대 행복도 (1번과 연결해서 갱신)
- 3) 맥주를 2개 고를 수 있는 서로 다른 모든 경우에 대해 날개로 1개 고른 뒤,
남은 돈과 남은 맥주들로 4캔 만원 행사를 적용하여 맥주를 구매할 때 얻을 수 있는 최대 행복도 (계속 연결해서 갱신)
- 4) 맥주를 3개 고를 수 있는 서로 다른 모든 경우에 대해 날개로 1개 고른 뒤,
남은 돈과 남은 맥주들로 4캔 만원 행사를 적용하여 맥주를 구매할 때 얻을 수 있는 최대 행복도 (계속 연결해서 갱신)

또한 날개 맥주가 0개, 1개, 2개, 3개일 때 각각에 대해서 시간복잡도는 $O(N)$, $O(N^2)$, $O(N^3)$, $O(N^4)$ 가 되며, 이에 총 시간복잡도는 $O(N^4)$ 이 됩니다.

현재 N 은 최대 100이므로 N^4 는 10^8 , 즉 1억이어서 주어진 시간 제한 내에 풀이가 가능합니다.

풀이를 정리하자면, 먼저 입력받은 맥주 목록을 행복도 순서대로 정렬한 뒤, 각 가능한 경우 4가지를 차례로 모두 확인하여 얻게 된 행복도가 최대인 때의 행복도를 출력하면 됩니다. 4가지 경우를 확인할 때, 각 날개 맥주를 선택하고 나서 4캔 만원 맥주를 고를 때에는 정렬해둔 목록을 이용하여 행복도가 큰 맥주부터 선택하여야 합니다.

정해 코드

```
#include <bits/stdc++.h>
using namespace std;
int N, M;
pair<int,int> beer[105];
int check[105];

int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);

    cin >> N >> M;
    for(int i=0; i<N; i++){
        cin >> beer[i].second >> beer[i].first;
    }
    int ans = 0;
```

```

sort(beer, beer+N);

/* case 1) 날개 0개 구입 */
{
    int money = M;        // 남은 돈
    int h = 0;            // 현재까지 구입한 맥주로 얻을 수 있는 총 행복도
    int temp = h;         // 4캔 만원 행사를 고려할 때 필요한 변수, 행복도
    int cnt = 0;          // 구입한 맥주 개수 카운터
    for(int j=N-1; j>=0; j--){
        if(money>=2500 && check[j]==0){        // 돈이 2500원 이상 남았고 인덱스 j인 맥
주를 아직 구입하지 않은 경우라면
            // 구입 !
            money -= 2500;
            cnt++;
            h += beer[j].first;
            // 이때 4캔 만원 행사를 고려하여 4개 단위로 확인
            if(cnt%4 ==0)
                temp = h;
        }
    }
    ans = max(ans, temp);
}

/* case 2) 날개 1개 구입 */
for(int i=0; i<N; i++){
    // 맥주 하나 날개로 선택 (4캔 만원 x)
    int money = M - beer[i].second;
    if(money < 0) continue;        // 남은 돈이 없으면 다음 맥주 확인
    check[i] = 1;

    int h = beer[i].first;
    int temp = h;
    int cnt = 0;
    for(int j=N-1; j>=0; j--){
        if(money>=2500 && check[j]==0){        // 돈이 2500원 이상 남았고 인덱스 j인 맥
주를 아직 구입하지 않은 경우라면
            // 구입 !
            money -= 2500;
            cnt++;
            h += beer[j].first;
            // 이때 4캔 만원 행사를 고려하여 4개 단위로 확인
            if(cnt%4 ==0)
                temp = h;
        }
    }
    ans = max(ans, temp);
    check[i] = 0;
}

/* case 3) 날개 2개 구입 */
for(int i=0; i<N; i++){
    for(int k=i+1; k<N; k++){
        // 맥주 두개 날개로 선택 (4캔 만원 x)
        int money = M - beer[i].second - beer[k].second;
        if(money < 0) continue;
        check[i] = check[k] = 1;

        int h = beer[i].first + beer[k].first;

```

```

        int temp = h;
        int cnt = 0;
        for(int j=N-1; j>=0; j--){
            if(money>=2500 && check[j]==0){          // 돈이 2500원 이상 남았고 인덱스 j
인 맥주를 아직 구입하지 않은 경우라면
                // 구입 !
                money -= 2500;
                cnt++;
                h += beer[j].first;
                // 이때 4캔 만원 행사를 고려하여 4개 단위로 확인
                if(cnt%4 ==0)
                    temp = h;
            }
        }
        ans = max(ans, temp);
        check[i] = check[k] = 0;
    }
}

/* case 4) 날개 3개 구입 */
for(int i=0; i<N; i++){
    for(int k=i+1; k<N; k++){
        for(int t=k+1; t<N; t++){
            // 맥주 세개 날개로 구입 (4캔 만원 x)
            int money = M - beer[i].second - beer[k].second - beer[t].second;
            if(money < 0) continue;
            check[i] = check[k] = check[t] = 1;

            int h = beer[i].first + beer[k].first + beer[t].first;
            int temp = h;
            int cnt = 0;
            for(int j=N-1; j>=0; j--){
                if(money>=2500 && check[j]==0){          // 돈이 2500원 이상 남았고 인
덱스 j인 맥주를 아직 구입하지 않은 경우라면
                    // 구입 !
                    money -= 2500;
                    cnt++;
                    h += beer[j].first;
                    // 이때 4캔 만원 행사를 고려하여 4개 단위로 확인
                    if(cnt%4 ==0)
                        temp = h;
                }
            }
            ans = max(ans, temp);
            check[i] = check[j] = check[t] = 0;
        }
    }
}

cout << ans;
return 0;
}

```