BAEKJOON – 포도주시식(https://www.acmicpc.net/problem/2156)

문제

일렬로 놓여진 포도주를 마시는 데, 연속해서 3잔을 마실 수 없다.

이러한 상황에서 최대한 많이 마실 수 있는 포도주의 양을 구하라.

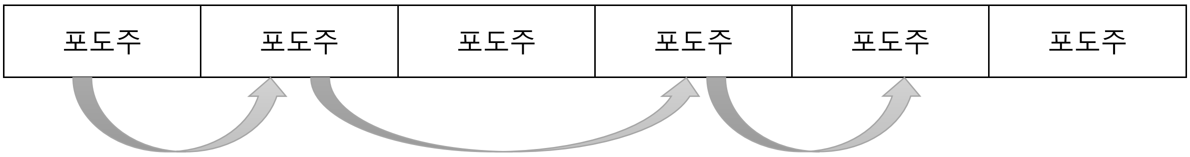
풀이

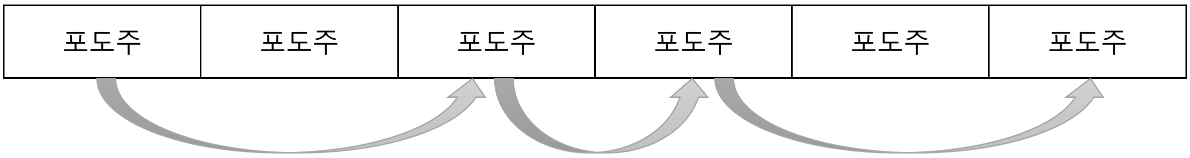
이 문제 다이나믹 프로그래밍으로 풀 수 있는 문제이다.

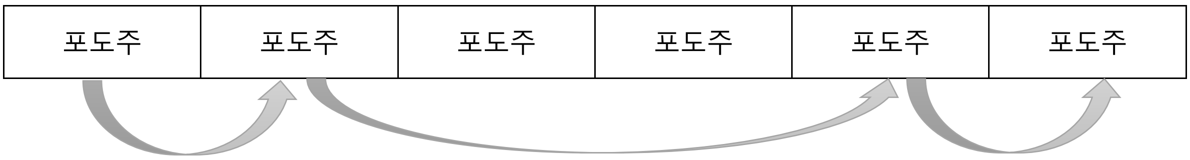
포도주	포도주	포도주	포도주	포도주	포도주

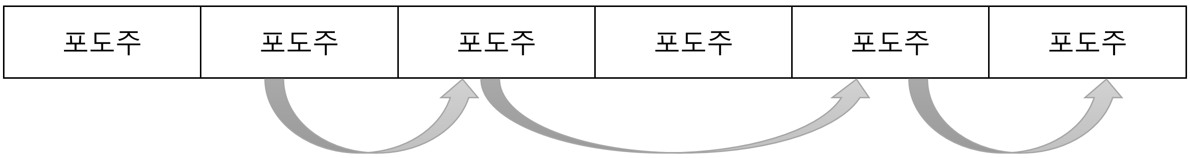
위의 6잔의 포도주에서 마실 수 있는 경우를 살펴보자.

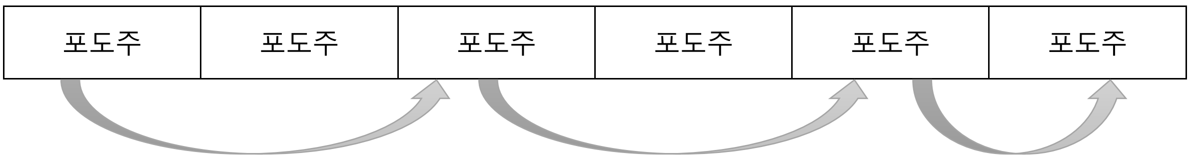
양의 고려하지 않고 생각해봤을 때, 다음과 같은 경로가 존재한다.











위의 경로를 보면 6잔 중에서 4잔은 꼭 마실 수 있다.(그림이 모든 경로가 아닐 수 있다.)

중요한 점은 포도주를 마실 수 있는 방법이 3가지인 것이다.

1. 나란히 붙어 있는 잔을 연속으로 마시는 경우
2. 한 잔을 마시지 않고 마시는 경우
3. 두 잔을 마시지 않고 마시는 경우

이렇게 3가지의 경우가 있을 때, 고려해야 할 경우는 1번이다.

이전 단계에서 1번을 선택한 경우, 다음 동작으로는 2번과 3번 밖에 할 수 없다.

이 고려사항을 생각하면서 코드를 작성하면 답을 구할 수 있다.

코드를 작성하기 위해서는 우선 와인의 양을 담을 배열이 필요하고 각 경우마다 maximum 값을 가지는 배열이 필요하다.

와인의 수를 cnt라고 할 때,

와인의 양 : wine[cnt]

각 단계의 최대 값 : total[cnt][3]을 만들 수 있고

total[cnt][0]은 경우 1번, total[cnt][1]은 경우 2번, total[cnt][2]은 경우 3번이라고 하자.

i번째의 와인을 마실 때, total은 다음과 같이 구할 수 있다.

total[i][0] = maximum(total[i-1][1], total[i-1][2]) + wine[i]

total[i][1] = maximum(total[i-2][0], total[i-2][1], total[i-2][2]) + wine[i]

total[i][2] = maximum(total[i-3][0], total[i-3][1], total[i-3][2]) + wine[i]

처음 3단계에 대한 total값은 따로 구해주고 cnt까지 반복하면 모든 total을 구할 수 있다.

마지막으로 주의할 점은 최대 값을 구할 때,

total[cnt][0], total[cnt][0], total[cnt][2] 이 세 값 중 최대 값을 구하는 것이 아니라

total[cnt-1][0]을 포함하여 최대 값을 구해야한다.