

04. Dynamic Programming 2

돌 게임

BOJ 9655번

두 명에서 게임을 진행한다.

각자 번갈아가며 돌을 1개 또는 3개 가져갈 수 있다.

마지막 돌을 가져가는 사람이 게임을 이기게 된다.

두 사람이 완벽하게 게임을 했을 때, 이기는 사람을 구하는 문제

돌 게임

BOJ 9655번

게임 이론

보통 두 사람이 게임을 하는데, 최대한 완벽하게 게임을 진행하게 된다.



돌 게임

BOJ 9655번

게임 이론

반드시 선공이 누군지 문제에 명시되어 있다!

무조건 선공이 유리하다.



선방필승!

돌 게임

BOJ 9655번

풀이 1. 다이나믹 프로그래밍

$dp[i]$: N이 i 일 때, 선공이 이기면 1, 지면 2

$dp[3] = 1$, N이 3이라면, 어떠한 경우에도 선공이 이길 수 있다!

돌 게임

BOJ 9655번

풀이 1. 다이나믹 프로그래밍

$N = 2$?

돌이 2개 있는데, 3개는 못가져가니 1개만 가져간다.

남은 1개는 후공이 가져가므로, $dp[2] = 2$

돌 게임

BOJ 9655번

풀이 1. 다이나믹 프로그래밍

$N = 4$?

4가 되려면 1 또는 3개 남았을 때 돌을 가져가면 된다!

i	1	2	3	4
dp[i]	1	2	1	

돌 게임

BOJ 9655번

풀이 1. 다이나믹 프로그래밍

반대로 생각해서, 자기가 가질 수 있는 만큼 다 가져가 본다.

다음 상황에서 자신이 승리할 수 있는 방법이 하나라도 존재하면 이길 수 있다!

i	1	2	3	4	5	6	7	8	9	10
dp[i]	1	2	1	2						

돌 게임

BOJ 9655번

풀이 1. 다이나믹 프로그래밍

점화식

$dp[i] = 1$ (if, $dp[i-1] = 2$ or $dp[i-3] = 2$)

$dp[i] = 2$ (else)

돌 게임

BOJ 9655번

풀이 2. 수학

각자 갖고갈 수 있는 돌의 개수 : 1, 3 개 (홀수)

어차피 선공은 홀수, 후공은 짝수개의 돌이 남았을 때 마지막 돌을 가져간다.



참고

돌 게임 2 ~ 6 (검색하면 나와요!)

가장 긴 증가하는 부분 수열

BOJ 11053번

어떤 수열이 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 문제

크기 N (1 ~ 1000)

가장 긴 증가하는 부분 수열

BOJ 11053번

부분 수열, 문자열 문제

연속인지(Substring), 연속이 아닌지(Subsequence) 확인

Substring [10, 20, 10, 30, 20, 40]

Subsequence [10, 20, 10, 30, 20, 40]

가장 긴 증가하는 부분 수열

BOJ 11053번

수열 $A = \{10, 20, 10, 30, 20, 50\}$

$dp[i]$: $A[i]$ 를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
 $A[i]$ 가 반드시 포함되어야 한다.

A	10	20	10	30	20	50
dp[i]						

가장 긴 증가하는 부분 수열

BOJ 11053번

수열 $A = \{10, 20, 10, 30, 20, 50\}$

$dp[i]$: $A[i]$ 를 마지막으로 하는 가장 긴 증가하는 부분 수열의 길이
 $A[i]$ 가 반드시 포함되어야 한다.

A	10	20	10	30	20	50	40
dp[i]	1	2	1	3	2	4	

가장 긴 증가하는 부분 수열

BOJ 11053번

점화식

$$dp[i] = \max(dp[0] \sim dp[i-1]) + 1 \quad (A[0 \sim i-1] < A[i])$$

정답

$dp[0 \sim n-1]$ 중 가장 큰 수

참고

전깃줄 (2565번)



lcs

BOJ 9251번

LCS(Longest Common Subsequence, 최장 공통 부분 수열)

두 수열이 주어졌을 때, 모두의 **부분 수열이 되는 수열 중 가장 긴 것을 찾는 문제**이다.

lcs

BOJ 9251번

lcs 구하는 방법

1. 두 단어 중 하나를 고정시켜 둔다.
2. 고정되지 않은 단어를 첫 글자부터 차례차례 고정된 단어와 비교를 하는데,
글자가 일치하면 그 전의 결과값에 +1 해주고, 일치하지 않다면 그 전 결과값 중 큰 값을 넣어준다.

lcs

BOJ 9251번

pringles angeles

두 문자열의 lcs 찾기

lcs

BOJ 9251번

	p	r	i	n	g	l	e	s
a								
n								
g								
e								
l								
e								
s								

lcs

BOJ 9251번

숫자의 변동은 두 가지로 인해 이루어질 수 있다.

1. 문자가 일치할 경우
2. 문자가 불일치하나, 바로 위 또는 바로 왼쪽 값의 최댓값이 0이 아닌 경우

lcs

BOJ 9251번

점화식

1. 문자가 일치할 경우 $\rightarrow \text{if}(A[i] == B[j]) \ d[i][j] = d[i-1][j-1] + 1$
2. 문자가 불일치하나, 바로 위 또는 바로 왼쪽 값의 최댓값이 0이 아닌 경우
 $\rightarrow \text{if}(A[i] != B[j]) \ d[i][j] = \max(d[i][j-1], d[i-1][j])$

참고

[lcs2 \(9252번\)](#)

평범한 배낭

BOJ 12865번

N개의 물건이 있고, 각 물건은 무게 W 와 가치 V 를 갖는다.

가방의 최대 무게 K 를 넘지 않는 선에서 최대 가치를 찾는 문제

평범한 배낭

BOJ 12865번

완전 유명한 냅색 문제!

dp table 의 공간을 어떻게 잡아야 할까? 메모리를 넘지 않는 선에서 겹치지 않게!

입력

첫 줄에 물품의 수 N ($1 \leq N \leq 100$)과 준서가 버틸 수 있는 무게 K ($1 \leq K \leq 100,000$)가 주어진다. 두 번째 줄부터 N 개의 줄에 걸쳐 각 물건의 무게 W ($1 \leq W \leq 100,000$)와 해당 물건의 가치 V ($0 \leq V \leq 1,000$)가 주어진다.

입력으로 주어지는 모든 수는 정수이다.

출력

한 줄에 배낭에 넣을 수 있는 물건들의 가치합의 최댓값을 출력한다.

평범한 배낭

BOJ 12865번

$dp[i][j]$: i 번째 물건까지 고려했고, 가방에 넣은 물건들의 무게가 j 일 때 최대 가치합

-> N 이 최대 100, W 가 최대 10만이므로 충분!

평범한 배낭

BOJ 12865번

$dp[i][j]$: i 번째 물건까지 고려했고, 가방에 넣은 물건들의 무게가 j 일 때 최대 가치합

1 ~ N 번째 물건까지 순회하며 가방에 담는다

i 번째 물건을 가방에 넣지 않는 경우 $\rightarrow dp[i][j] = dp[i-1][j]$

i 번째 물건을 가방에 넣는 경우 $\rightarrow dp[i][j] = dp[i-1][j-w[i]] + v[i]$

평범한 배낭

BOJ 12865번

```
//1 ~ n 번째 물건 돌아가며 탐색
for (int i = 1; i <= n; i++) {
    //가방 무게
    for (int j = 1; j <= k; j++) {
        if (j >= w[i]) {
            dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - w[i]] + v[i]);
        }
        else dp[i][j] = dp[i - 1][j];
        ans = max(ans, dp[i][j]);
    }
}
```

파이프 옮기기 1

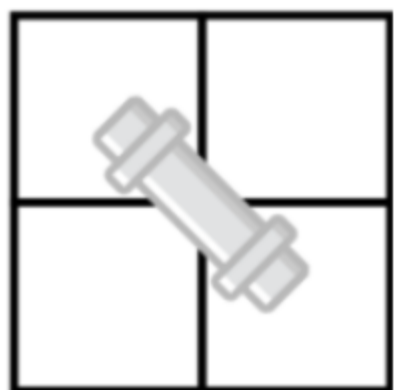
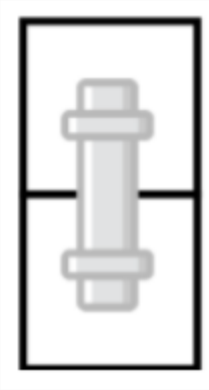
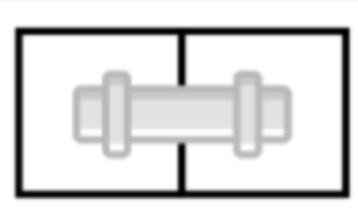
BOJ 17070번

삼성 A형 기출 문제

$(1, 1) \sim (n, n)$ 까지 파이프를 옮기는 방법의 개수를 구하는 문제

파이프 옮기기 1

BOJ 17070번



파이프 옮기기 1

BOJ 17070번

$dp[r][c][k]$: (r, c) 칸에서 k 모양 파이프로 끝나는 경우의 수

$dp[r][c][0]$: $dp[r][c-1][0] + dp[r][c-1][2]$

$dp[r][c][1]$: $dp[r-1][c][1] + dp[r-1][c][2]$

$dp[r][c][2]$: $dp[r-1][c-1][0 \sim 2]$ (단, $(r-1, c), (r, c-1)$ 칸에 벽이 있으면 안됨!)

징검다리

2016 SCPC 예선

<https://www.codeground.org/>

삼성 프로그래밍 경진대회 문제들을 풀어 볼 수 있어요!

징검다리

2016 SCPC 예선

N개의 돌로 이루어진 징검다리를 건너려 한다.

최대 K칸 떨어진 돌로 이동할 수 있다 (1~k 칸 모두 이동 가능)

지뢰가 놓여있는 돌은 밟을 수 없다.

연속으로 같은 수의 돌을 건널 수 없다.

-> N번째 돌을 밟으면 다리를 건널 수 있을 때, 다리를 건너는 가짓수는?

징검다리

2016 SCPC 예선

$dp[i][j]$: 가장 최근 j 칸 이동해 i 번째 돌로 도착한 경우의 수

최근 j 칸에 대한 기록은 다음에 j 칸 이동을 막아주는 역할을 한다.

$$dp[i][j] = \sum_{x=1}^k dp[i-j][x] \text{ (단, } x \neq j \text{)}$$

징검다리

2016 SCPC 예선

코드

```
for (int i = 1; i <= n; i++)
{
    //지뢰가 있다면 볼 필요 없음
    if (check[i]) continue;
    for (int j = 1; j <= k; j++)
    {
        if (i - j < 0) break;
        else if (check[i - j]) continue;
        else
        {
            dp[i][j] = (psum[i - j] - dp[i - j][j] + MOD) % MOD;
            psum[i] += dp[i][j] % MOD;
        }
    }
}
```