

극장 좌석

<https://www.acmicpc.net/problem/2302>

- $dp[i]$ = 좌석이 i 개일 때 앉을 수 있는 경우의 수 (단, VIP 석 제외)
- $dp[i] = dp[i-1] + dp[i-2]$

| | | | | | |
|---|---|-------|-------|-------|-----|
| 1 | 2 | | $i-2$ | $i-1$ | i |
|---|---|-------|-------|-------|-----|

- 위와 같이 새로운 좌석 i 가 추가되면

① i 가 i 번째 좌석에 앉는 경우 : $dp[i-1]$

| | | | | | |
|---|---|-------|-------|-------|-----|
| 1 | 2 | | $i-2$ | $i-1$ | i |
|---|---|-------|-------|-------|-----|

② i 가 $i-1$ 번째 좌석에 앉는 경우 : $dp[i-2]$

| | | | | | |
|---|---|-------|-------|-----|-------|
| 1 | 2 | | $i-2$ | i | $i-1$ |
|---|---|-------|-------|-----|-------|

- $dp[i] = dp[i-1] + dp[i-2]$
- 초깃값 : $dp[0] = 1, dp[1] = 1$
- VIP 석이 추가 되면 VIP 석 앞 되를 기준으로 나누어서 생각하고 결과 값을 곱하면 된다.

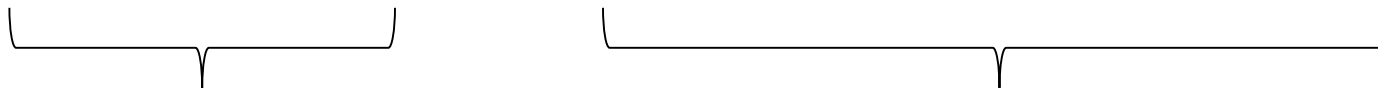
| | | | | | | |
|---|---|-----|---|---|---|---|
| 1 | 2 | VIP | 4 | 5 | 6 | 7 |
|---|---|-----|---|---|---|---|

극장 좌석

<https://www.acmicpc.net/problem/2302>

- VIP 석이 추가 되면 VIP 석 앞 뒤를 기준으로 나누어서 생각하고 결과 값을 곱하면 된다.

| | | | | | | |
|---|---|--------|---|---|---|---|
| 1 | 2 | 3(VIP) | 4 | 5 | 6 | 7 |
|---|---|--------|---|---|---|---|



$$dp[3 - 0 - 1] = dp[2]$$

$$dp[(7+1) - 3 - 1] = dp[4]$$

- VIP 석을 저장한 배열을 A라고 하였을 때 가장 처음에 0번과 가장 마지막에 N+1을 저장하면 $A[i] - A[i-1] - 1$ 로 구간을 일반화 할 수 있다.

ex) $3 - 0 - 1 = 2$, $(7+1) - 3 - 1 = 4$

```
11 solve(int i){  
    if (i == 0) return 1;  
    if (i == 1) return 1;  
  
    if (dp[i] != -1) return dp[i];  
    dp[i] = 0;  
    return dp[i] += solve(i - 1) + solve(i - 2);  
}
```

```
11 ans = 1;  
for (int i = 1; i < A.size(); ++i) {  
    ans *= solve(A[i] - A[i - 1] - 1);  
}
```