

계단 수

<https://www.acmicpc.net/problem/1562>

- 쉬운 계단 수 문제를 이용하여 계단 수 문제를 풀어본다.
- 쉬운 계단 수 문제 : $dp[i][j]$ = 길이가 i 이고 마지막 숫자가 j 인 계단 수의 개수
- $dp[i][j] = dp[i-1][j-1] + dp[i-1][j+1]$
- 계단 수 문제 : $dp[i][j][s]$ = 길이가 i 이고 마지막 숫자가 j , 현재 등장한 숫자 집합 s 일 때
- 계단 수의 경우의 수 (현재 j 는 집합 s 에 속한다.)
 - ① j 가 이미 등장한 경우 : $dp[i-1][j+1][s] + dp[i-1][j-1][s]$
 - ② j 가 처음으로 등장한 경우 : $dp[i-1][j+1][s \& \sim(1 \ll k)] + dp[i-1][j-1][s \& \sim(1 \ll k)]$
- $dp[i][j][s] = ① + ②$
 $= dp[i-1][j+1][s] + dp[i-1][j+1][s] +$
 $dp[i-1][j+1][s \& \sim(1 \ll k)] + dp[i-1][j+1][s \& \sim(1 \ll k)]$

계단 수

<https://www.acmicpc.net/problem/1562>

```
ll solve(int i, int j, int s) {
    if (i == 1) {
        if (1 <= j && j <= 9) {
            if (s == (1 << j)) return 1LL;
            else return 0;
        }
        else {
            return 0;
        }
    }

    if (dp[i][j][s] != -1) return dp[i][j][s];

    dp[i][j][s] = 0;
    if (s & (1 << j)) {
        if (j + 1 <= 9) dp[i][j][s] += solve(i - 1, j + 1, s); dp[i][j][s] %= mod;
        if (j - 1 >= 0) dp[i][j][s] += solve(i - 1, j - 1, s); dp[i][j][s] %= mod;
        if (j + 1 <= 9) dp[i][j][s] += solve(i - 1, j + 1, s & ~(1 << j)); dp[i][j][s] %= mod;
        if (j - 1 >= 0) dp[i][j][s] += solve(i - 1, j - 1, s & ~(1 << j)); dp[i][j][s] %= mod;
    }
    return dp[i][j][s];
}
```