

자물쇠

<https://www.acmicpc.net/problem/1514>

- 각각의 디스크마다 그 디스크를 가장 왼쪽으로 하면서 연속으로 한 칸, 두 칸, 세 칸을 몇 바퀴나 돌릴 지 결정해야 한다.
- $dp[i][j][k]$ = i 번째 디스크를 돌릴 것이고($i \rightarrow N$), 현재 칸은 j 만큼 돌아가 있고, 다음 칸은 k 만큼 돌아가 있는 상태
- 현재 i 번째 디스크를 가리키고 있는 값 : $(A[i] + j) \% 10$
 - ☞ i 번째 디스크를 기준으로 연속 두 칸(two) / 세 칸(three) 만큼 돌린다.
 i 번째 디스크는 $(A[i] + j + two + three) \% 10$ 을 가리키고 있다.
- 한 칸(one) 돌리는 양은 목적지 숫자에 해당하는 $B[i] - (A[i] + j + two + three) \% 10$ 만약 음수가 나온다면 10을 더해주어 순방향으로 돌릴 시 얼마만큼 돌려야 할지 변환한다.
 - ☞ turn 배열 = {0, 1, 1, 1, 2, 2, 2, 1, 1, 1}
 - : 순방향 기준으로 자물쇠를 돌릴 시 몇 번 만에 돌릴 수 있는지 값을 저장
 - ex) turn[1] : 1 만큼 돌려야 하면 1번 만에 돌릴 수 있다.
turn[4] : 4만큼 돌려야 하면 2번 만에 돌릴 수 있다. (한번에 최대 3만큼 가능)
turn[7] : 7만큼 돌려야 하면 반대 방향으로 1번 만에 돌릴 수 있다.

자물쇠

<https://www.acmicpc.net/problem/1514>

- $dp[i][j][k] = \min(\text{turn}[\text{one}] + \text{turn}[\text{two}] + \text{turn}[\text{three}] + dp[i+1][(k + \text{two} + \text{three}) \% 10][\text{three}])$
- two, three : [0, 10), one : two/three 값에 따라 정해짐
- 초깃값 : $dp[N] == 0$

```
int solve(int i, int j, int k) {
    if (i == N) return 0;

    if (dp[i][j][k] != -1) return dp[i][j][k];
    dp[i][j][k] = 1e9;

    int origin = (A[i] + j) % 10;
    int to = B[i];

    for (int three = 0; three < 10; ++three) {
        for (int two = 0; two < 10; ++two) {
            int from = (origin + two + three) % 10;
            int one = to - from;
            if (one < 0) one += 10;
            int cost = turn[one] + turn[two] + turn[three];
            dp[i][j][k] = min(dp[i][j][k], cost + solve(i + 1, (k + two + three) % 10, three));
        }
    }
    return dp[i][j][k];
}
```