

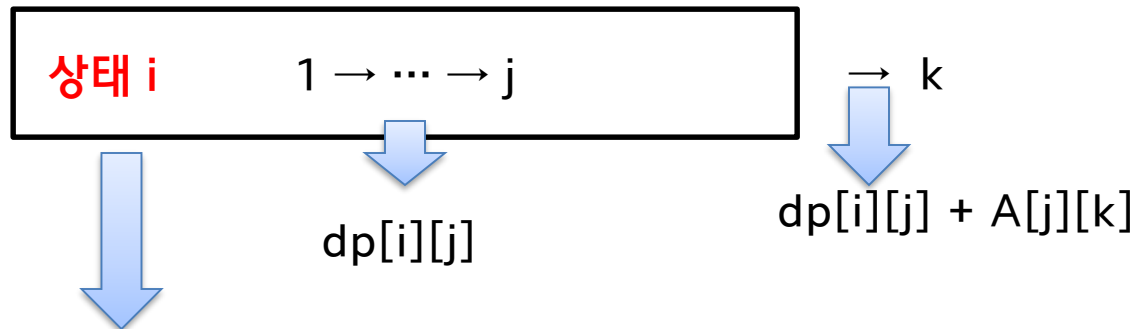
외판원 순회

<https://www.acmicpc.net/problem/2098>

- $dp[i][j]$ = 도시를 방문한 상태가 i 이고, 마지막 방문 도시가 j 일 때 최소값
- 시작 도시는 1로 고정한다. ($1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ 과 $2 \rightarrow 3 \rightarrow 1 \rightarrow 2$ 는 같다.)
- $1 \rightarrow 4$ 까지 정답이 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ 라고 하면 $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ 는 방문한 집합은 같기 때문에 추후에 고려할 필요가 없는 값이 된다.

즉, $1 \rightarrow 5$ 로 갈 때 4에서 가는 경우라면

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ 가 $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5$ 보다 무조건 최소값이 된다.

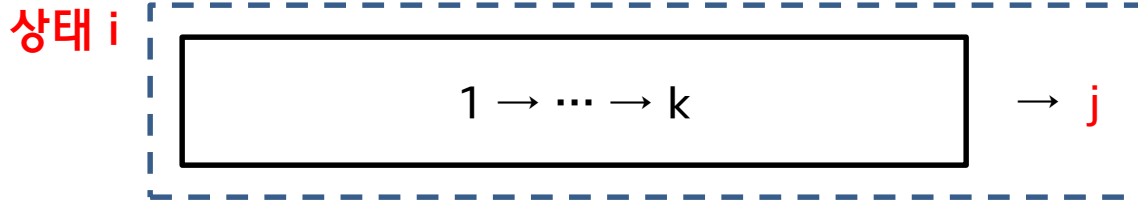


- $dp[i | (1 \ll k)][k]$ 로 갱신할 수 있다.
 - ① $i \& (1 \ll j) \neq 0$: j 가 집합 i 에 들어가 있어야 한다.
 - ② $i \& (1 \ll k) == 0$: k 가 집합 i 에 없어야 한다.
 - ③ $A[j][k] \neq 0$: $j \rightarrow k$ 로 이동가능해야 한다.

외판원 순회

<https://www.acmicpc.net/problem/2098>

- $dp[i][j]$ = 도시를 방문한 상태가 i 이고, 마지막 방문 도시가 j 일 때 최소값



- $dp[i][j] = \min(dp[s \& \sim(1 \ll j)][k] + A[k][j])$
 - ① $i \& (1 \ll k) \neq 0$: k 가 집합 i 에 속한다.
 - ② $i \& (1 \ll j) \neq 0$: j 가 집합 i 에 속한다.
 - ③ $A[k][j] \neq 0$: k 와 j 는 연결되어 있다.
- 초기값 : $dp[(1 \ll 0)][0] = 0$, 나머지 MAX
- 정답 : $\min(dp[(1 \ll N) - 1][i] + A[i][0])$ ($i=0 \sim N-1$)

외판원 순회

<https://www.acmicpc.net/problem/2098>

```
dp[(1 << 0)][0] = 0;
for (int i = 0; i < (1 << N); ++i) {
    for (int j = 1; j < N; ++j) {
        for (int k = 0; k < N; ++k) {
            if (j != k && (i & (1 << j)) && (i & (1 << k) && w[k][j])) {
                dp[i][j] = min(dp[i][j], dp[i & ~(1 << j)][k] + w[k][j]);
            }
        }
    }
}
```

```
int solve(int i, int j) {
    if (i == (1 << 0) && j == 0) return 0;

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

    dp[i][j] = 1e9;
    for (int k = 0; k < N; ++k) {
        if (j != k && (i & (1 << j)) && (i & (1 << k)) && w[k][j]) {
            dp[i][j] = min(dp[i][j], solve(i & ~(1 << j), k) + w[k][j]);
        }
    }
    return dp[i][j];
}
```

