외판원 순회

```
dp = vii((1 << N) + 10, vi((20), 1e9));
dp[(1 << 0)][0] = 0;
for (int i = 0; i < (1 << N); ++i) {
        for (int j = 0; j < N; ++j) {
                for (int k = 0; k < N; ++k) {
                        if (j != k && (i & (1 << j)) != 0 && (i & (1 << k)) == 0 && w[j][k] != 0) {
                                dp[i | (1 << k)][k] = min(dp[i | (1 << k)][k], dp[i][j] + w[j][k]);
                        }
                }
        }
}


dp = vii((1 << N) + 10, vi((20), 1e9));
dp[(1 << 0)][0] = 0;
for (int i = 0; i < (1 << N); ++i) {
        for (int j = 0; 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]);
                        }
                }
        }
}
```

외판원 순회

```cpp
dp = vii((1 << N) + 10, vi((20), 1e9));
dp[(1 << 0)][0] = 0;
for (int i = 0; i < (1 << N); ++i) {
        for (int j = 0; j < N; ++j) {
                for (int k = 0; k < N; ++k) {
                        if (j != k && (i & (1 << j)) != 0 && (i & (1 << k)) == 0 && w[j][k] != 0) {
                                dp[i | (1 << k)][k] = min(dp[i | (1 << k)][k], dp[i][j] + w[j][k]);
                        }
                }
        }
}


dp = vii((1 << N) + 10, vi((20), 1e9));
dp[(1 << 0)][0] = 0;
for (int i = 0; i < (1 << N); ++i) {
        for (int j = 0; 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]);
                        }
                }
        }
}
```

Gaussian

외판원 순회

```
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];
}
```

Gaussian