```cpp
/*** 1119 - Pimp My Ride:
Given Individual jobs, are numbered 1 through n. Given the base price
p for each job and a surcharge s for every pair of jobs (i, j),
meaning that you have to pay additional s for job i,
if and only if job j was completed before,
you are to compute the minimum total costs needed to finish all jobs.

Input starts with test case an integer T (≤100).
Each case starts with an integer n (1 ≤ n ≤ 14) denoting number of jobs.
Then follow n lines, each containing exactly n integers.
The I'th line contains the surcharges that have to be paid in garage number
i for the I'th job and the base price for job i.
More precisely, on the I'th line, the I'th integer is the base price
for job i and the j'th integer i ≠ j is the surcharge for job i that
applies if job j has been done before.
The prices will be non-negative integers smaller than or equal to 100000.

For each case, print the case number and the minimum total cost.
****/
#include<bits/stdc++.h>
using namespace std;
int check(int mask,int pos){ return mask & (1<<pos); }
int seton(int mask,int pos){ return mask | (1<<pos); }
int setoff(int mask,int pos) { return mask & (~(1 << pos)); }
int settoggle(int mask,int pos) { return mask ^ (1 << pos); }
int setToX(int mask,int pos,int x){return mask ^ ((-x^mask)&(1<<pos));}
/// setToX()-> pos'th bit will be seton if x is 1, and set off if x is 0.

int n,dp[1<<15],a[25][25];
int fun(int mask){
    if(mask == (1<<n)-1)return 0;

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

    int mn = 1<<28;
    for(int i=0; i<n; i++){
        if(check(mask,i)==0){
            int cost = a[i][i];
            for(int j=0; j<n; j++){
                if(i!=j && check(mask,j)!=0){
                    cost += a[i][j];
                }
            }
            int ret = cost + fun(seton(mask,i));
            mn = min(mn,ret);
        }
    }
    return dp[mask] = mn;
}
int main(){
    int tt; scanf("%d",&tt);
    for(int ks=1; ks<=tt; ks++){
        scanf("%d",&n);
        for(int i=0; i<n; i++){
            for(int j=0; j<n; j++){
                scanf("%d",&a[i][j]);
            }
        }

        memset(dp,-1,sizeof(dp));
        int ans = fun(0);
        printf("Case %d: %d\n",ks,ans);
    }
    return 0;
}
```