**Problem Statement**

Let there be N workers and N jobs. Any worker can be assigned to perform any job, incurring some cost that may vary depending on the work-job assignment. It is required to perform all jobs by assigning exactly one worker to each job and exactly one job to each agent in such a way that the total cost of the assignment is minimized.

**Input Format**

Number of workers and job: N

Cost matrix C with dimension N\*N where C(i,j) is the cost incurred on assigning ith Person to jth Job.

**Sample Input**

4

[

9 2 7 8

6 4 3 7

5 8 1 8

7 6 9 4

]

**Sample Output**

13

**Constraints**

N <= 20

Solution:

#include <bits/stdc++.h>

using namespace std;

int dp[21][1<<21];

int cost[21][21];

int solve(int i,int mask,int n)

{

if(i==n)

{

return 0;

}

if(dp[i][mask]!=-1)

{

return dp[i][mask];

}

int ans=INT\_MAX;

for(int j=0;j<n;j++)

{

if((1<<j)& mask)continue;

mask=mask|(1<<j);

ans=min(ans,cost[j][i]+solve(i+1,mask,n));

mask=mask^(1<<j);

}

return dp[i][mask]=ans;

}

void input()

{

int n;

cin>>n;

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

cin>>cost[i][j];

}

}

memset(dp,-1,sizeof(dp));

cout<<solve(0,0,n)<<endl;

}

int main() {

/\*\*

\* Write your C++ code from here

\*/

int t;

cin>>t;

while(t--)

{

input();

}

return 0;

}