tsp.cpp

```
/* FILE: stp.cpp   last change: 5-Feb-2014   author: Romeo Rizzi
 * an implementation for a 2^n DP algorithm to the STP problem.
 */

//#define NDEBUG   // NDEBUG definita nella versione che consegno
#include <cassert>
#ifndef NDEBUG
#  include <iostream>  // uso di cin e cout non consentito in versione finale
#endif
#include <fstream>
#include <algorithm>

using namespace std;

const int UNKNOWN = -1;
const int MAX_N = 20, POW_2_MAX_N = 1024*1024;  int n;
int C[MAX_N][MAX_N];
int opt_val[MAX_N][ POW_2_MAX_N ]; // int opt_choice[MAX_N][ POW_2_MAX_N ];

int mymax( int a, int b ) { return (a>b)? a : b; }
int is_on(int i, int throughSet) { return throughSet & (1 << i); }
void display(int throughSet) {
  for(int i = 0; i < n; i++)
    cout << ( ( is_on( i, throughSet) ) ? 1 : 0 ) << " ";
  cout << endl;
}
void turn_on(int i, int *throughSet) { *throughSet = *throughSet | (1 << i); }
void turn_off(int i, int *throughSet) { *throughSet = *throughSet ^ (1 << i); }

int maxPath(int from, int throughSet) {
  //  cout << "ENTER PROBLEM: from = " << from << endl; display(throughSet);
  if( opt_val[from][throughSet] != UNKNOWN )  return opt_val[from][throughSet];
  int best_so_far = C[from][n-1]; // opt_choice[from][throughSet] = n-1;
  for(int i = 1; i < n-1; i++)
    if( is_on(i, throughSet) ) {
      turn_off(i, &throughSet);
      best_so_far = mymax(
                          best_so_far,
                          C[from][i] + maxPath(i, throughSet)
                         );
      turn_on(i, &throughSet);
    }
  return opt_val[from][throughSet] = best_so_far;
}


int main() {
  ifstream fin("input.txt"); assert( fin );
  fin >> n;
  for(int i = 0; i < n; i++)
     for(int j = 0; j < n; j++)
       fin >> C[i][j];
  fin.close();
  for(int i = 0; i < n; i++)
     for(int j = 0; j < POW_2_MAX_N; j++)
       opt_val[i][j] = UNKNOWN;
  ofstream fout("output.txt"); assert( fout );
  int throughSet = 0;
  for(int i = 1; i < n-1; i++)
    turn_on(i, &throughSet);
  // display(throughSet);
  fout << maxPath(0, throughSet) << endl;
  // display(throughSet);
  fout.close();
  return 0;
}
```