Traveling Sales Person

CS 491 – Competitive Programming

Mattox Beckman

University of Illinois at Urbana-Champaign
Department of Computer Science

Spring 2023

Objectives

- Understand how to use DP to solve the TSP problem.
- Use the bitmask technique to represent the cities already visited.
- Know the limit of *n* for brute force and DP problems.

The Problem

- You are given a set of nodes with weighted edges
  - cities and cost / time for travel
- Want to make a *tour*, visit all cities, return to start.
- What is the cheapest way to do this?
- Time complexity
  - $\mathcal{O}(n!)$ — check each permutation
  - Fix first city to take advantage of symmetry gives $(n-1)!$ solutions.
  - In a contest, brute force check can work up to about $n = 11$, max.

Setup

```
int main() {
   cin >> n;
   vvi adj(n);

   for(i=0; i<n; ++i)
      for(j=0; j<n; ++j) {
         cin >> c;
         adj[i].push_back(c);
      }

   mx = (1 << n) - 1;
   cout << "Best path has cost " << tsp(adj,mx,0,1) << endl
}
```

Using DP

```
vvi dp(16,vi(65536)); int n;
int tsp(vvi &costs, int &mx, int cur, int state) {
   if (dp[cur][state]>0) return dp[cur][state];
   if (state == mx) return costs[cur][0]; // return home

   int minleft = INF;  int bit=2;

   for(int i=1; i<n; ++i)  {
     if ( (state & bit) == 0) { // i not visited
        minleft = min(minleft, costs[cur][i] +
                          tsp(costs,mx,i,state | bit));
     }
     bit <<= 1;
   }
   return dp[cur][state]=minleft;
}
```