Traveling salesman

Saturday, October 28, 2023 6:31 AM

Given a complete graph with weighted edges, what is the min cost Hamiltonian cycle.

Path that visits every node once

It is an NP-complete problem.

Brute force $O(n!)$ | Dynamic programming $O(2^n \cdot n^2)$ | Selling on ebay :) $O(1)$

Dynamic programming

- for computing paths of len = N use data for N-
- Store the optimal direct path from a node to every other node.

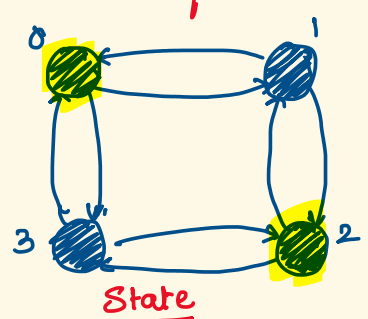
optimization.

State:

1. The nodes visited so far.
2. The index of the last visited node.

In total, there are $2^N \cdot N$ states.
space complexity.

We use a single 32-bit integer & use its bits to represent selected nodes



State
(0101, 2)
(4) ↑ last visited

Code:

```python
def TSP_recursive(prev, mask):
    if (prev, mask) in dp: return dp[(prev, mask)]
    if mask == (1 << n) - 1:
        return 0 ## all nodes visited

    cost = inf
    for i in range(n):
        if mask & (1 << n): continue
        cost = min(cost, dist[prev][i] + TSP(i, mask | (1 << n)))
    dp[(prev, mask)] = cost # memoize
    return cost

def TSP_iterative():
    dp = [[inf] * (1 << n) for _ in range(n)]

    for i in range(n): dp[i][(1 << i] = 0

    for mask in range(1, 1 << n):
        for i in range(n): # already included
            if mask & (1 << i):
                for j in range(n): # include a new node
                    if i != j and mask & (1 << j): continue
                    dp[j][mask | (1 << j)] = min(
                                                dp[j][mask | (1 << j)],
                                                dp[i][mask] + dist[i][j]
                                            )

    return min([
            x[(1 << n) - 1] for x in dp[i]
        ], default = inf)
```

n=3 $2^3-1=(7)_{10}=(111)_2$

cost till current node

Already visited

Track best cost to reach i..

Base case: Cost with only one node = 0

try all 2^n combinations

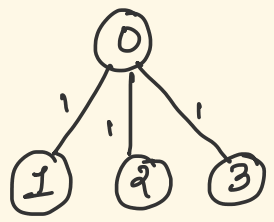
node i included

Notice that we use a push dp in the iterative code. This is because, for doing bottom up there is no way for us to know in advance, what the best cost is for mask (1<<n)-1.

Return the min cost for mask that includes all nodes ending at one of the nodes.

★ We need direct shortest path between every node pair in the graph.

eg. Consider a graph.

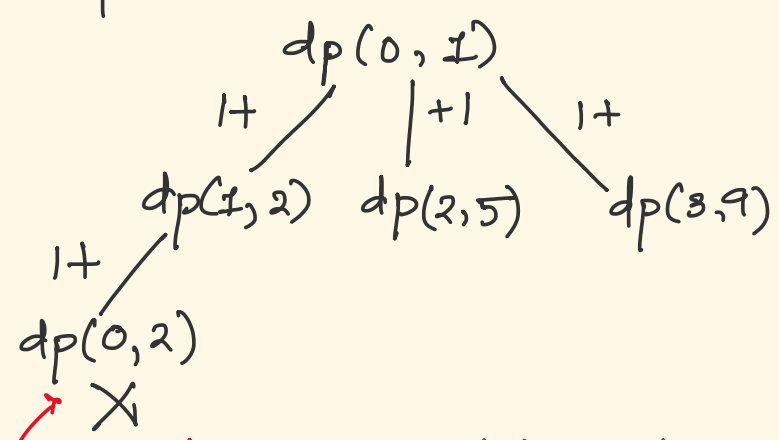


Therefore we must run floyd Warshall algo to get all pair shortest path.

If we are give adj list convert to adj mat & find APSP.

with state. dp(i, mask)

say we start with '0'.



we need to explore this route as we may reach other nodes.

But CANT! As the node is already in the mask.