# Design and Analysis of Algorithm (DAA)

## Dynamic Programming

# (Traveling Salesperson Problem)

Dr. Dayal Kumar Behera

School of Computer Engineering
KIIT Deemed to be University, Bhubaneswar, India

# Traveling Salesman Problem

Given a set of N cities and the distances between each pair, find the shortest possible route that visits each city exactly once and returns to the origin city.

(Find the shortest route visiting each city once, returning to the start)

**Applications:**

- Logistics
- Routing
- Genome sequencing
- Scheduling

# Basic Terminology

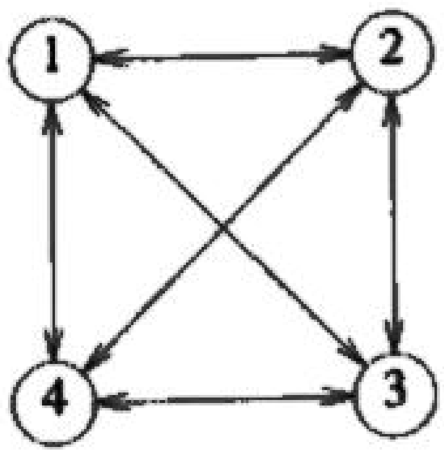
**Hamiltonian Cycle:** A path in a graph that visits each vertex exactly once and returns to the starting vertex.

**Types of TSP:**

- Symmetric TSP (STSP):
    - The distance from city A to city B is the same as from B to A.
- Asymmetric TSP (ATSP):
    - The distances from A to B and B to A can differ.

# Brute-Force Approach

The brute-force approach computes all possible permutations of cities to find the minimum tour distance.

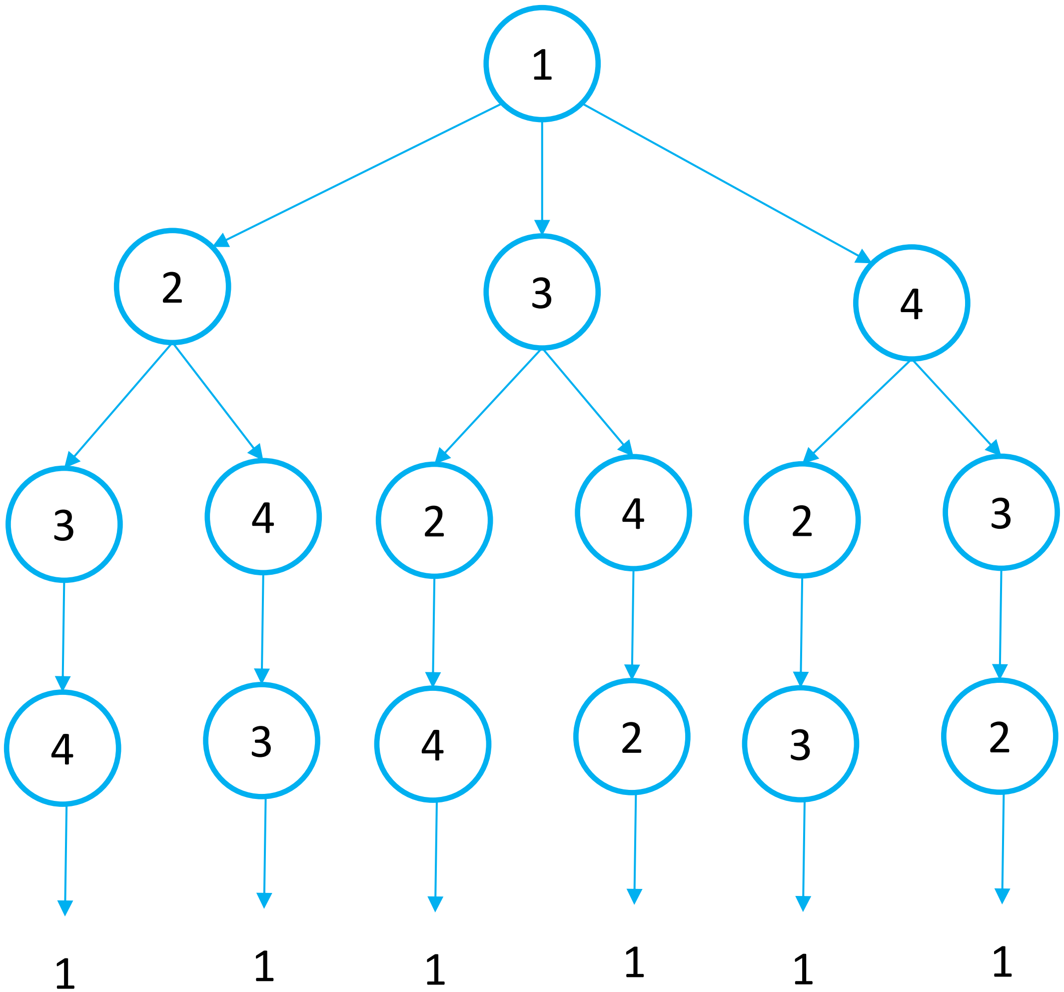



| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

**Time Complexity: O(N!), infeasible for large N.**

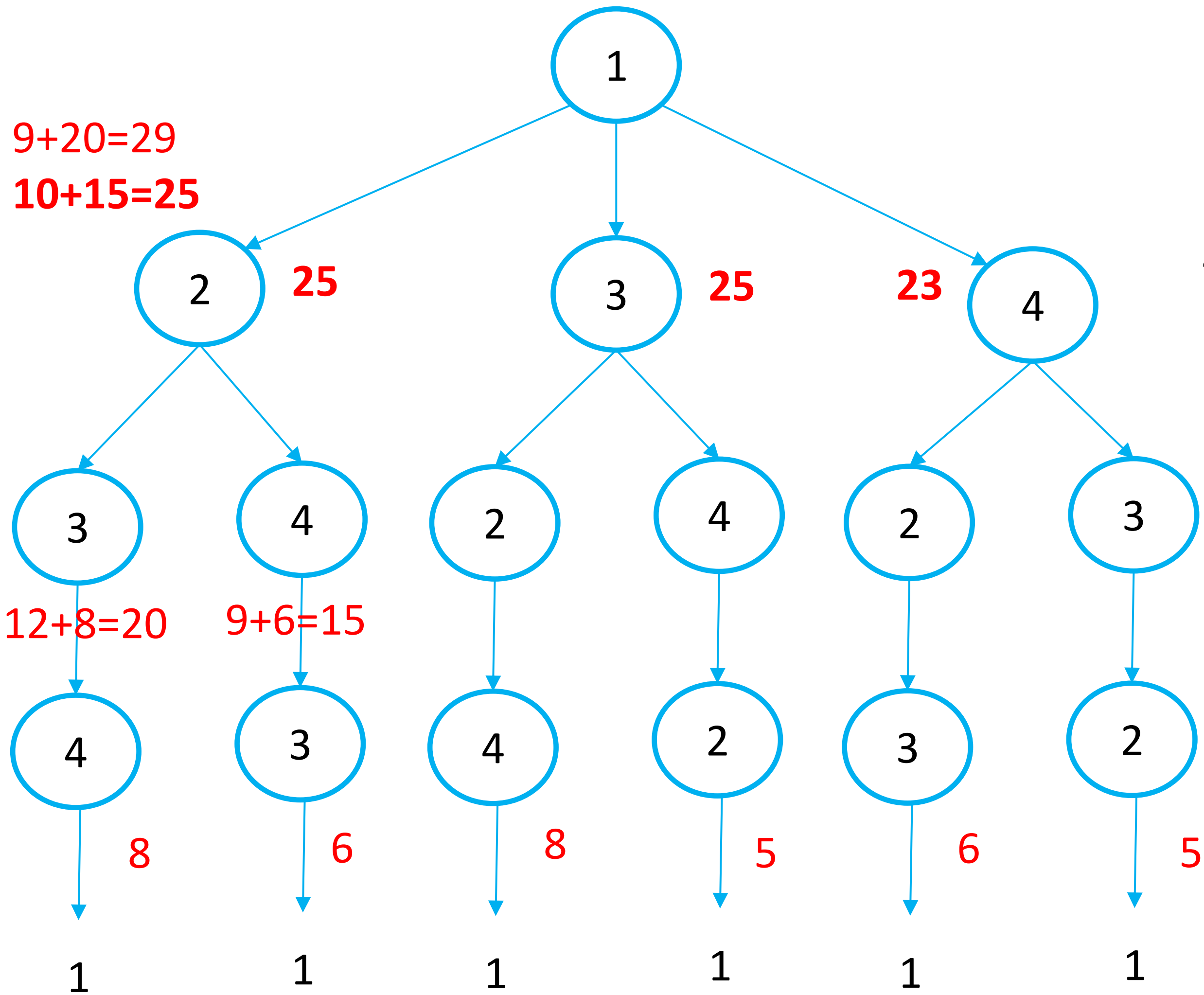
TSP is an NP-hard problem; thus, an efficient, exact solution does not exist for large inputs (i.e., large N).

# Brute-Force Approach



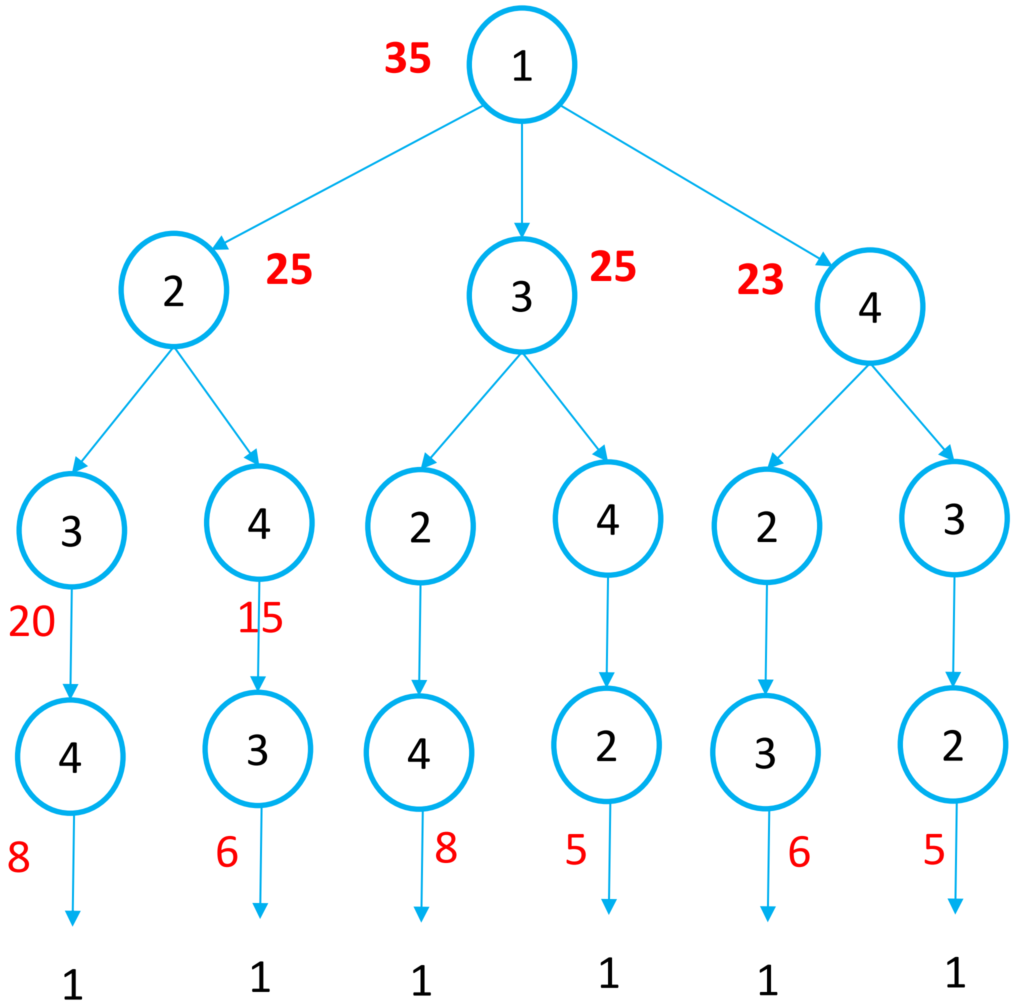

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

# Brute-Force Approach




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

# Brute-Force Approach




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

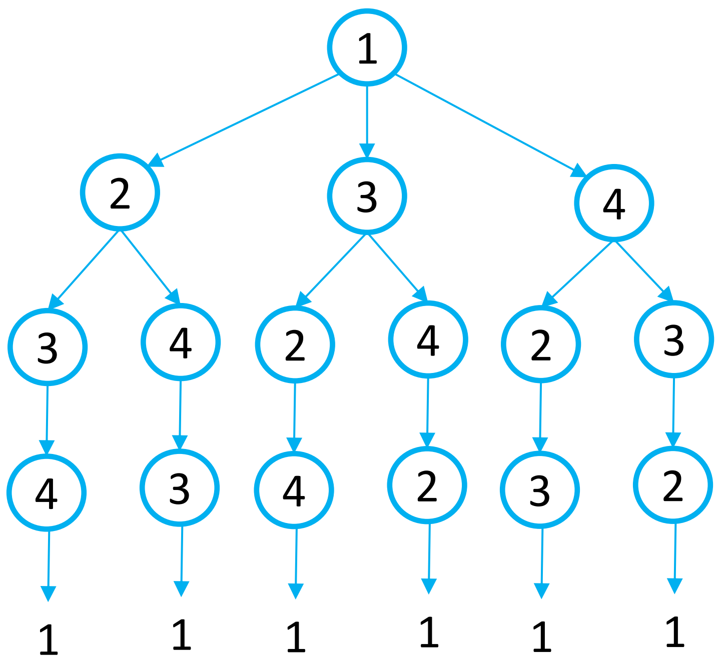
# TSP: Dynamic Programming

Use DP to break down the problem into sub-problems, solving smaller, overlapping sub-problems only once and storing their solutions.

# TSP: Recursive Relation

Let $g(i, S) =$ length of shortest path starting from vertex $i$ going through all vertices in $S$ and terminating at starting vertex (origin city).

$$g(1, \{2,3,4\}) = \min\{\, c_{12} + g(2, \{3,4\}), c_{13} + g(3, \{2,4\}), c_{14} + g(4, \{2,3\})\}$$




$g(1, \{2,3,4\}) = \min\{\, c_{1k} + g(k, \{3,4\})\}$
where $k \in \{2, 3, 4\}$

$$g(1, \{2,3,4\}) = \min_{k \in \{2,3,4\}} \{c_{1k} + g(k, \{3,4\})\}$$

$$g(i, S) = \min_{k \in S} \{c_{ik} + g(k, S - \{k\})\}$$

# TSP: Recursive Relation

Let $g(i, S) =$ length of shortest path starting from vertex $i$ going through all vertices in $S$ and terminating at starting vertex (origin city).

$g(1, V - \{1\})$ : length of minimum cost path

1: Starting and Returning vertex(city)
$V - \{1\}$ : Set of all vertices except 1

$$g(i, S) = \min_{k \in S}\{c_{ik} + g(k, S - \{k\})\} \quad \text{where } i \notin S$$

$g(k, V - \{1, k\}) =$ starting from vertex k covering all the vertices in $V - \{1, k\}$ and returning back to 1

# DP Approach

This is a bottom-up approach. So, we will first solve the base.

$g(1, \emptyset) = 0$ (Cost to stay at the starting city without travelling is zero)

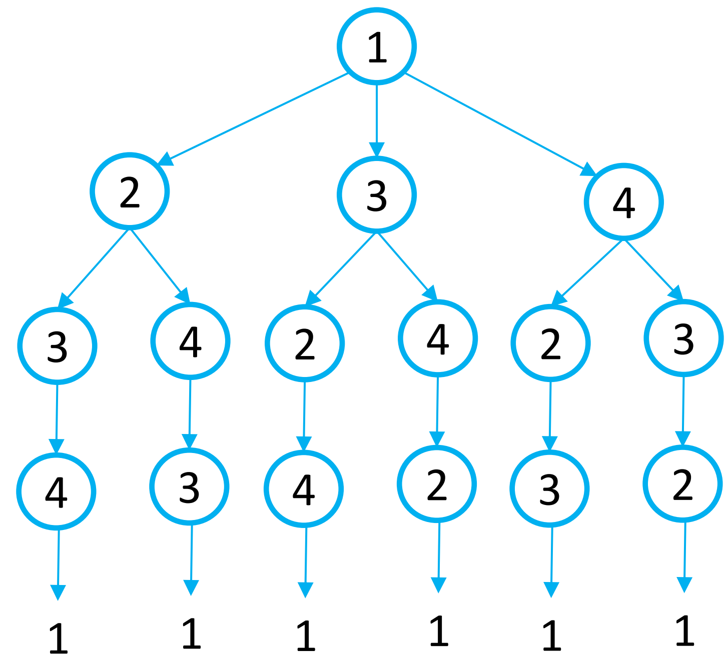
| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

$g(2, \emptyset)$ : Starting from vertex 2 covering no vertex and returning to 1

$g(2, \emptyset) = c_{21} = 5$

$g(3, \emptyset) = c_{31} = 6$

$g(4, \emptyset) = c_{41} = 8$



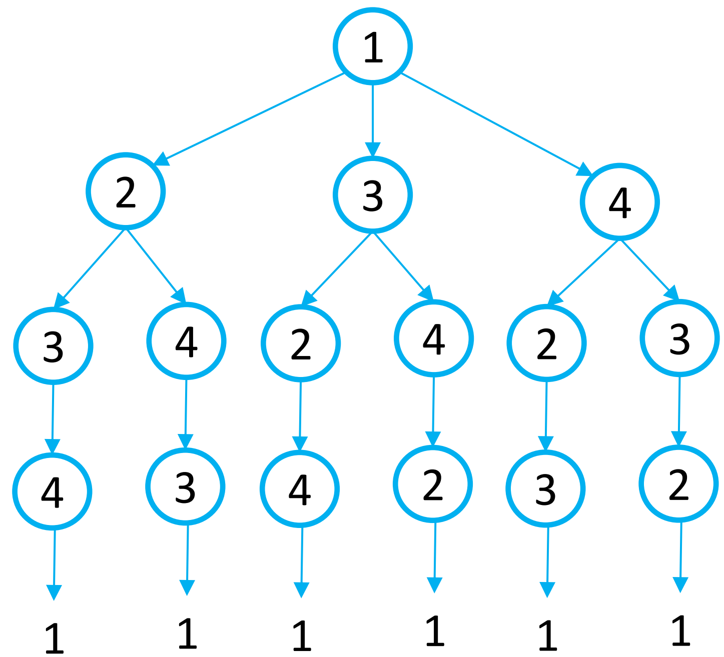
# DP Approach

$g(2,\{3\})$ : Starting from vertex 2, covering 3 and returning to 1

$$g(2,\{3\}) = c_{23} + g(3,\emptyset) = 9 + 6 = 15$$

$$g(2,\{4\}) = c_{24} + g(4,\emptyset) = 10 + 8 = 18$$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |



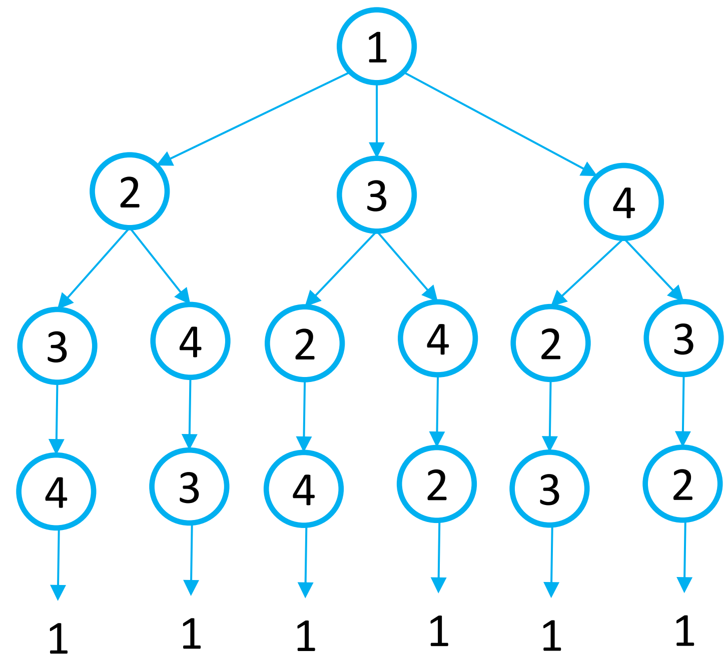
# DP Approach

$g(3, \{2\})$ : Starting from vertex 3, covering 2 and returning to 1

$$g(3, \{2\}) = c_{32} + g(2, \emptyset) = 13 + 5 = 18$$

$$g(3, \{4\}) = c_{34} + g(4, \emptyset) = 12 + 8 = 20$$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |



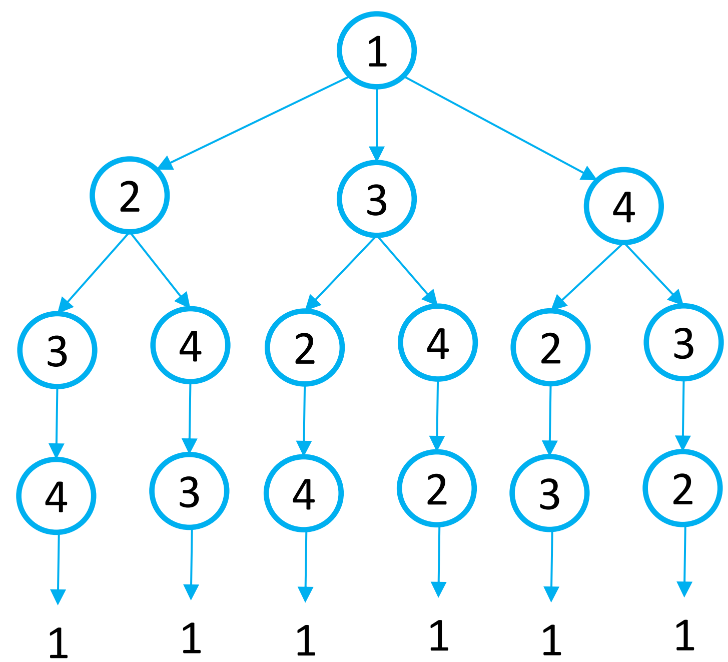
# DP Approach

$g(4,\{2\})$ : Starting from vertex 4, covering 2 and returning to 1

$$g(4,\{2\}) = c_{42} + g(2,\emptyset) = 8 + 5 = 13$$

$$g(4,\{3\}) = c_{43} + g(3,\emptyset) = 9 + 6 = 15$$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

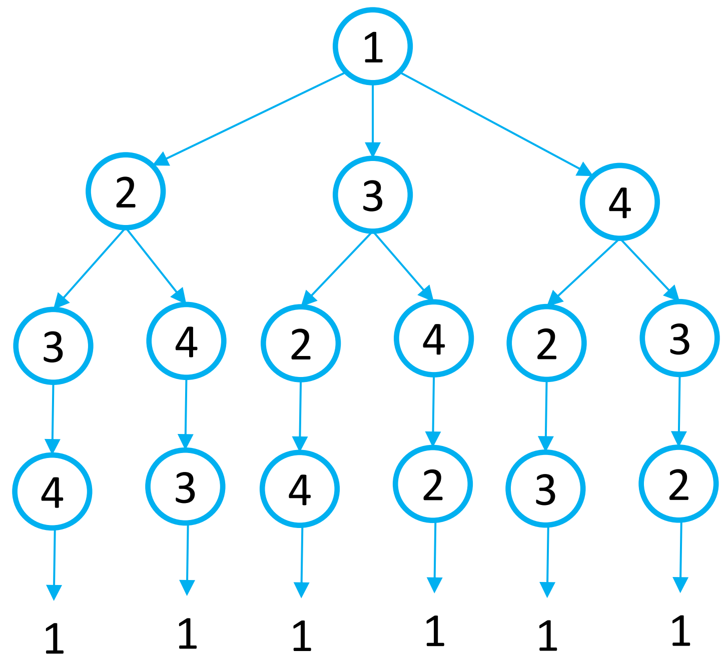


# DP Approach

$g(2,\{3,4\})$ : Starting from vertex 2, covering 3, 4 and returning to 1

$$g(2,\{3,4\}) = \min\{\, c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\}$$
$$= \min\{\, 9 + 20, 10 + 15\}$$
$$= 25$$

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |




$g(3,\{2,4\}) = 25$

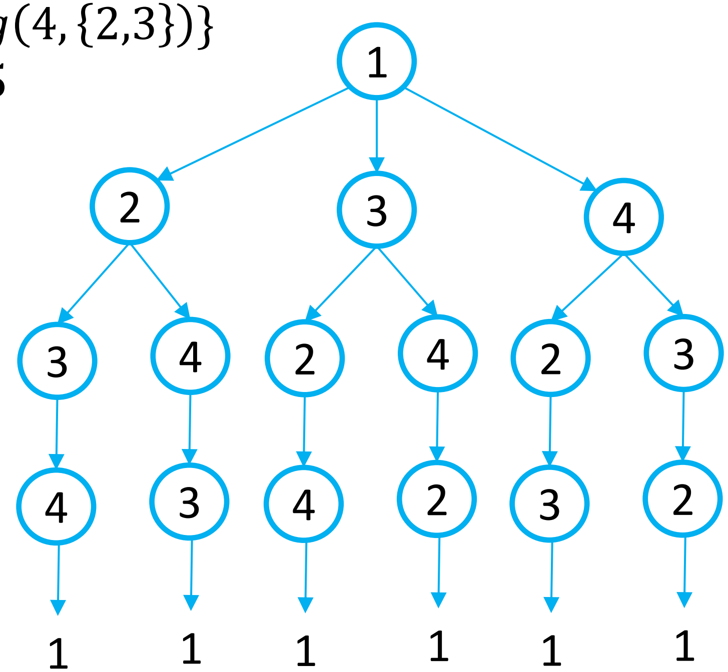
$g(4,\{2,3\}) = 23$

# DP Approach

$g(1,\{2,3,4\})$ : Starting from vertex 1, covering 2, 3, 4 and returning to 1

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 10 | 15 | 20 |
| 2 | 5 | 0 | 9 | 10 |
| 3 | 6 | 13 | 0 | 12 |
| 4 | 8 | 8 | 9 | 0 |

$$g(1,\{2,3,4\})$$
$$= \min\{\, c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\})\}$$
$$= \min\{\, 10 + 25, 15 + 25, 20 + 23\} = 35$$



# DP Table Construction

**Table dimensions:** $N \, x \, 2^N$ where $2^N$ represents all possible subsets of cities.

**Subset Encoding:** Use bit masking to encode subsets e.g. S = 1101, represents cities {1, 3, 4}

**Computation Order:** Bottom-up; start with smaller subsets and build up to larger ones.

# Complexity Analysis

Space Complexity: $O\ (N\ .\ 2^N)$

Time Complexity: $O\ (N^2\ .\ 2^N)$

To calculate the value of $g(i, S)$ we need to consider all cities in S. This results in O(N) work per state.

There are $O\ (N\ .\ 2^N)$ states and for each state, we compute the transition in $O\ (N\ )$ times.

Thus overall time complexity: $O\ (N\ .\ 2^N\ .N) =\ O\ (N^2\ .\ 2^N)$


Each of your
actions will
have an
impact on your
future.


Once you know
who is walking
with you on your path.
you will never
be afraid.

# Thank you