DESIGN & ANALYSIS OF ALGORITHM

PCC-CS402

DESIGN & ANALYSIS OF ALGORITHM SCHEDULE ----TOPIC WISE

	Topic	Sub Topic
1	INTRODUCTION	DESIGN OF ALGORITHM ,ANALYSIS OF ALGORITHM, ALGORITHM PROPERTIES
2	FRAMEWORK FOR ALGORITHM ANALYSIS	HOW TO COUNT EXECUTION TIME OF ALGORITHM,INPUT INSTANCES
3	ASYMPTOTIC NOTATION	BEST CASE,AVERAGE CASE, WORST CASE
4	SOLVING RECURRENCE RELATION	SUBSTITUTION METHOD, MASTER THEOREM
5	ALGORITHM DESIGN TECHNIQUES	DIVIDE & CONQUER, GREEDY,DYNAMIC PROGRAMMING, BACKTRACKING,
6	DISJOINT SET MANIPULATION	UNION FIND
7	NETWORK FLOW PROBLEM	FORD FULKERSON ALGORITHM
8	NP COMPLETENESS	NP,NP HARD.........ALGORITHM
9	APPROXIMATION ALGORITHM	COMPLEXITY ANALYSIS OF NP COMPETE PROBLEM

- **TRAVELLING SALESMAN PROBLEM**

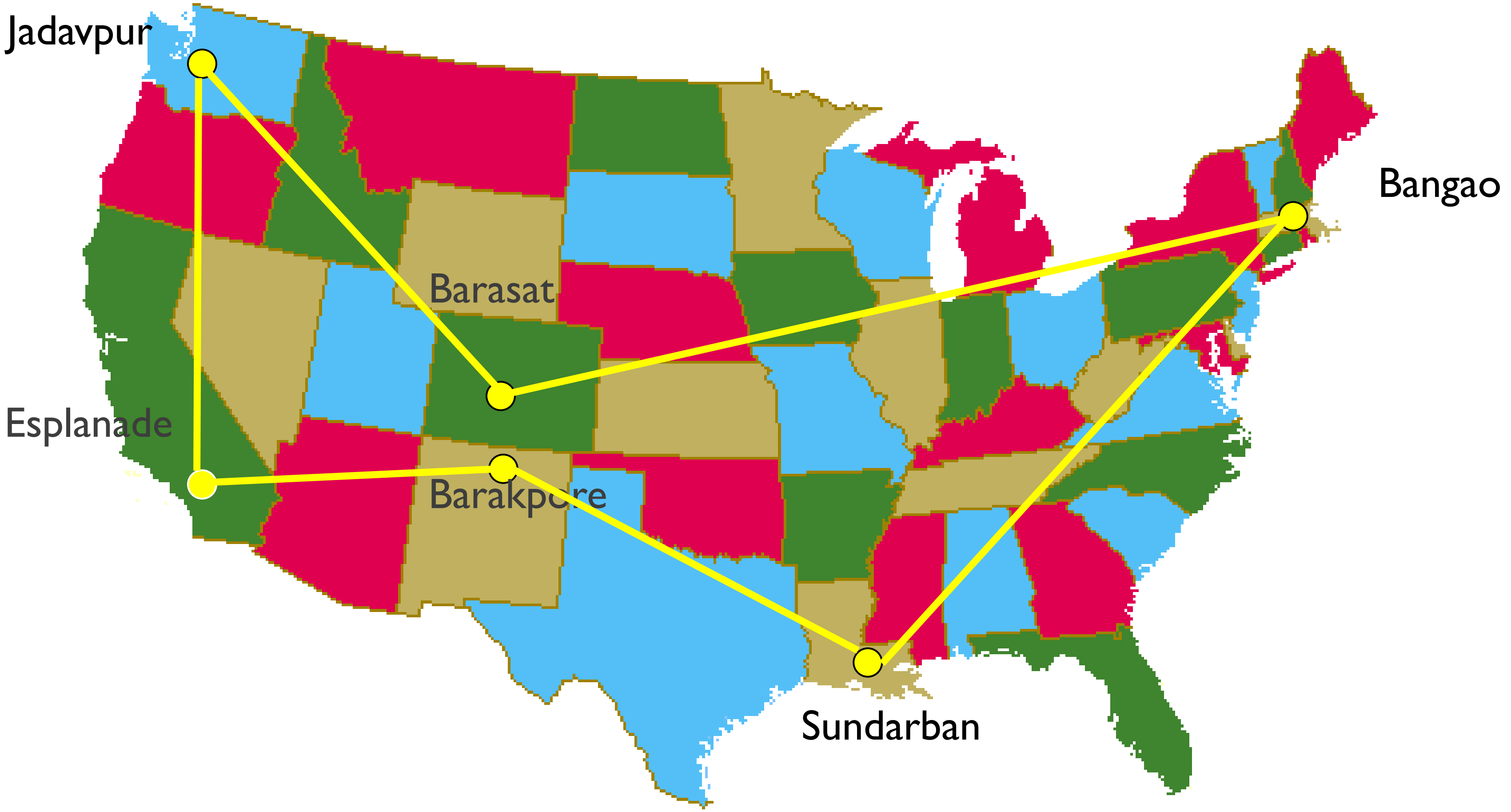
TRAVELING SALESMAN PROBLEM

- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- OUTCOME-

  Identify the smallest weight cycle.

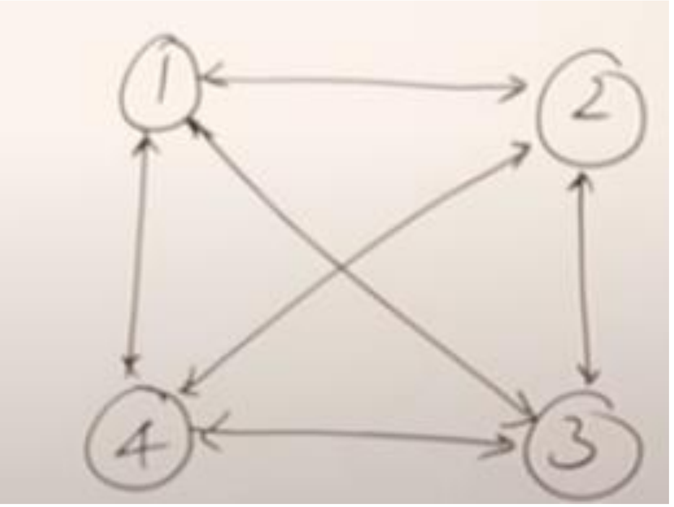
TRAVELING SALESMAN PROBLEM

6 CITIES -------- SHORTEST POSSIBLE ROUTE------------------EXACTLY ONCE



TRAVELING SALESMAN PROBLEM VS HAMILTONIAN CYCLE

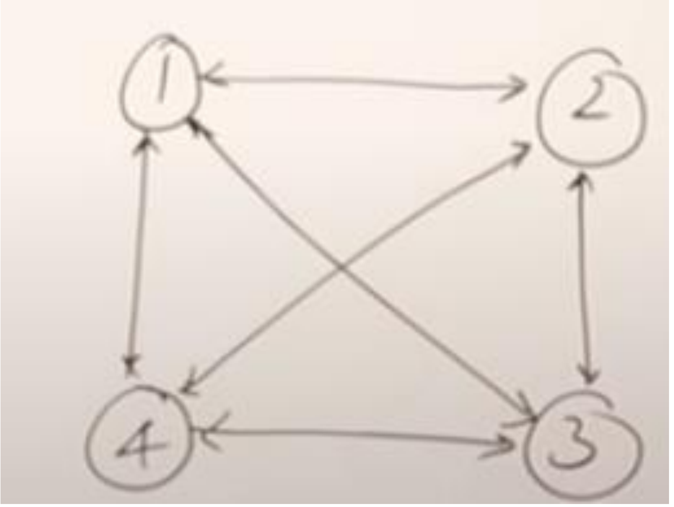
- Find a minimum weight Hamiltonian Cycle



APPLICATION TSP

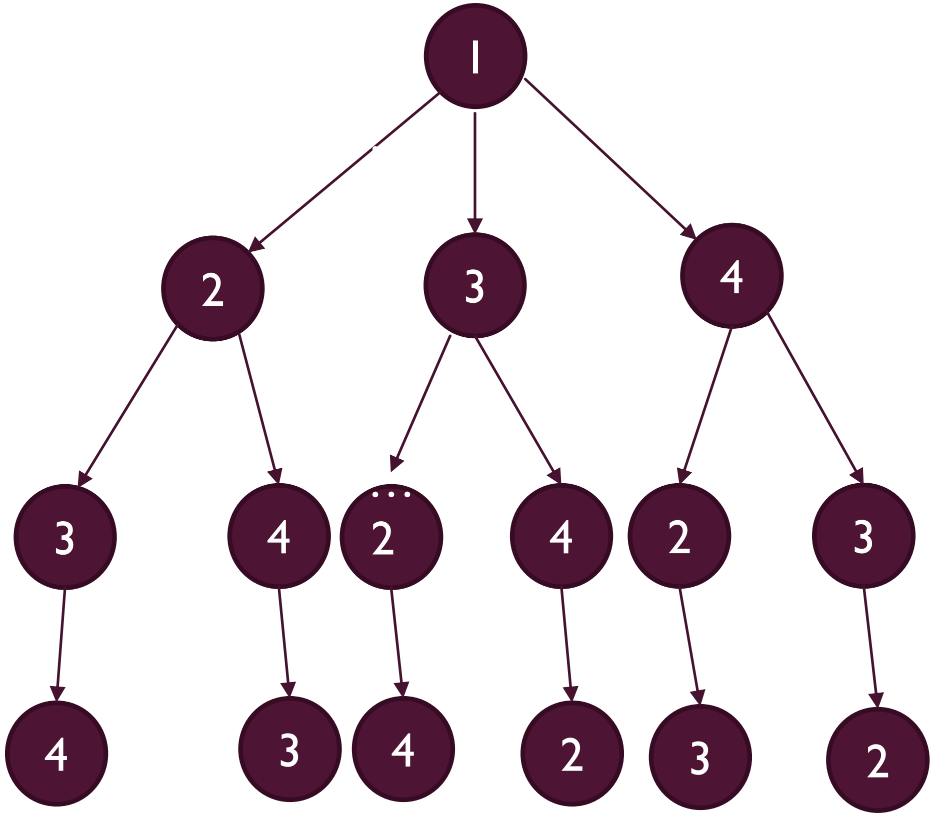
The TSP has many practical applications
-manufacturing
-plane routing
-telephone routing
-networks
-traveling salespeople
-structure of crystals

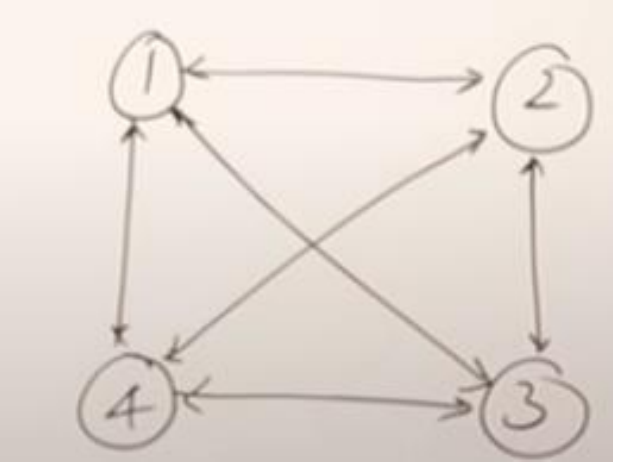
TRAVELING SALESMAN PROBLEM



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix} \end{array}$$

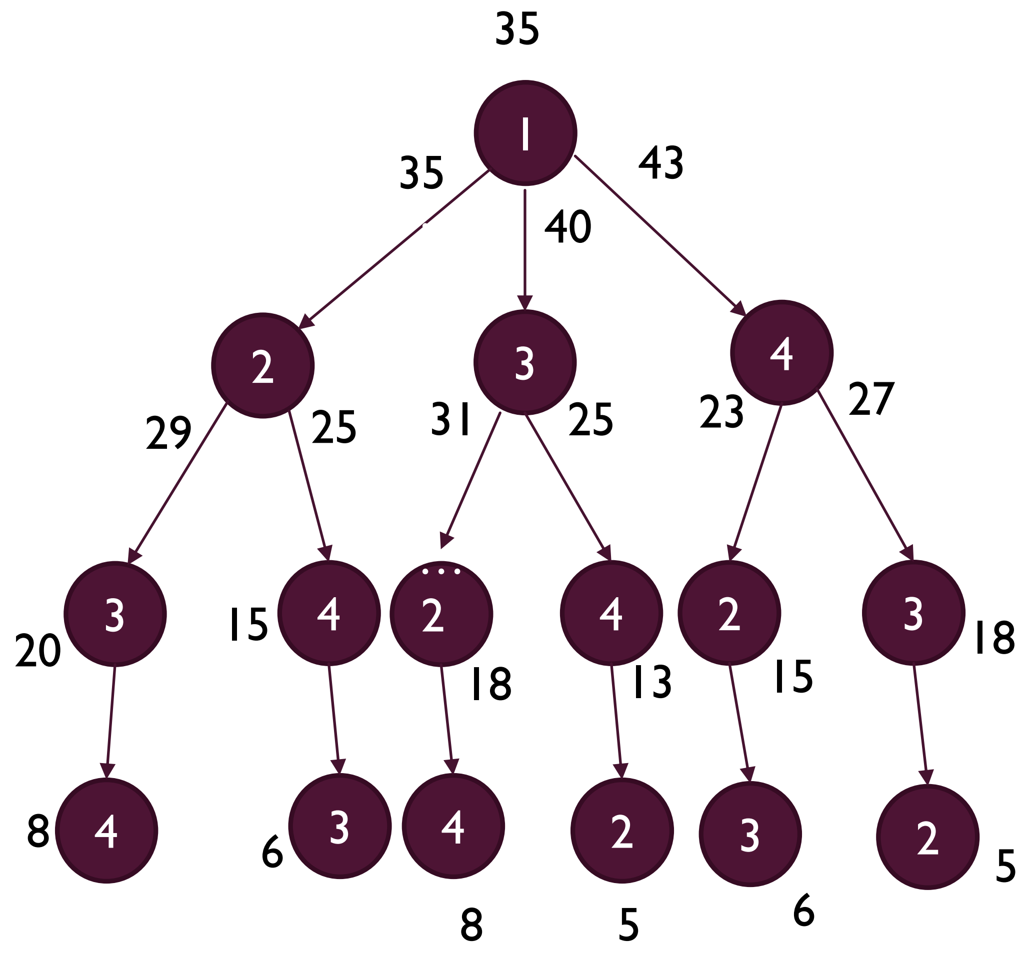
TRAVELING SALESMAN PROBLEM STATE SPACE TREE

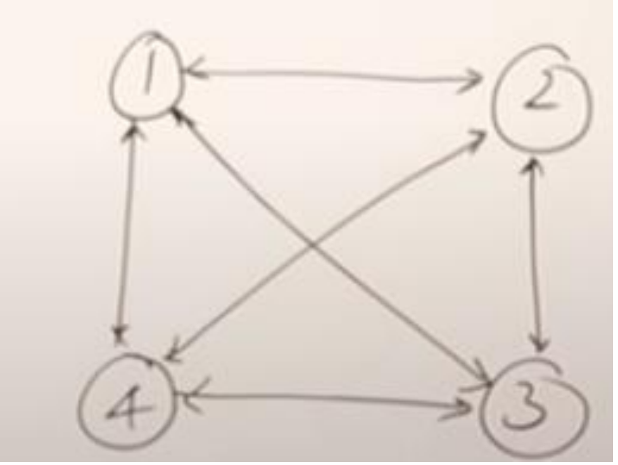




$$
A = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 10 & 15 & 20 \\ 2 & 5 & 0 & 9 & 10 \\ 3 & 6 & 13 & 0 & 12 \\ 4 & 8 & 8 & 9 & 0 \end{array}
$$

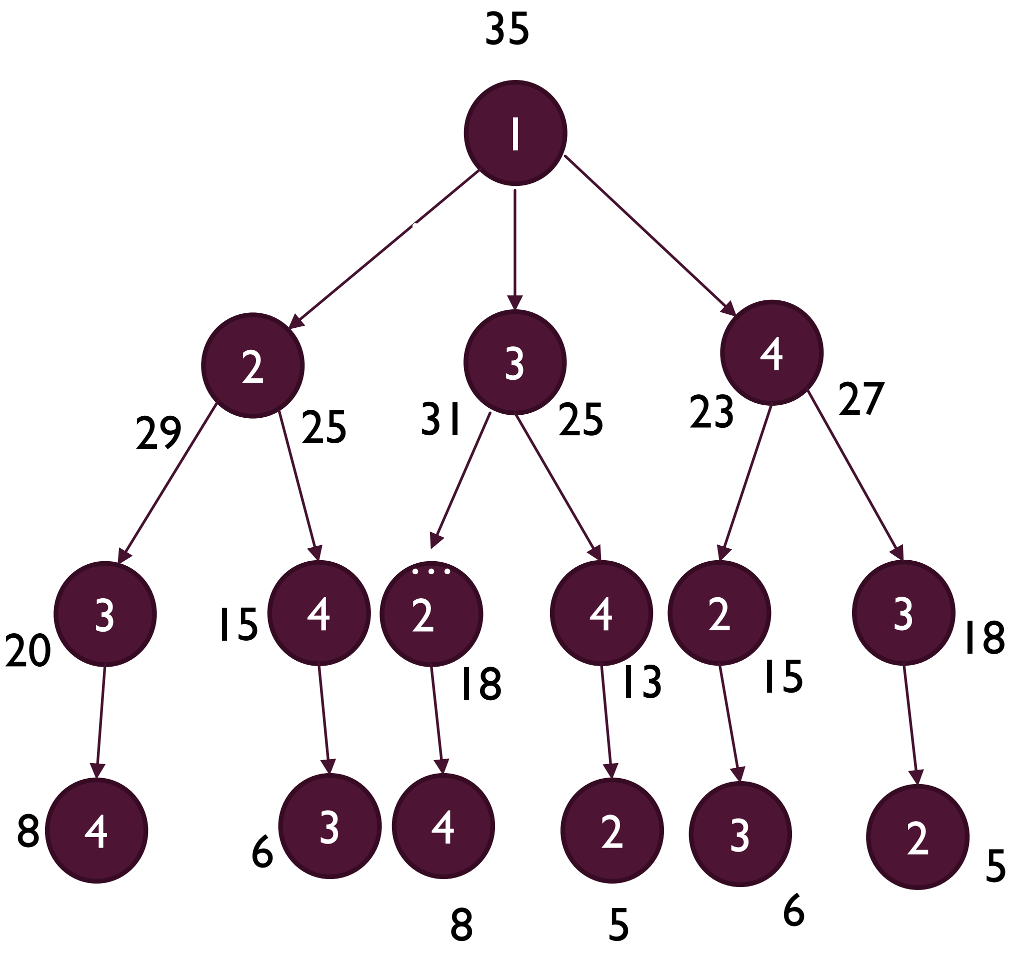
TRAVELING SALESMAN PROBLEM BRUTE FORCE APPROACH





$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix} \end{array}$$

TRAVELING SALESMAN PROBLEM
DYNAMIC PROGRAMMING APPROACH

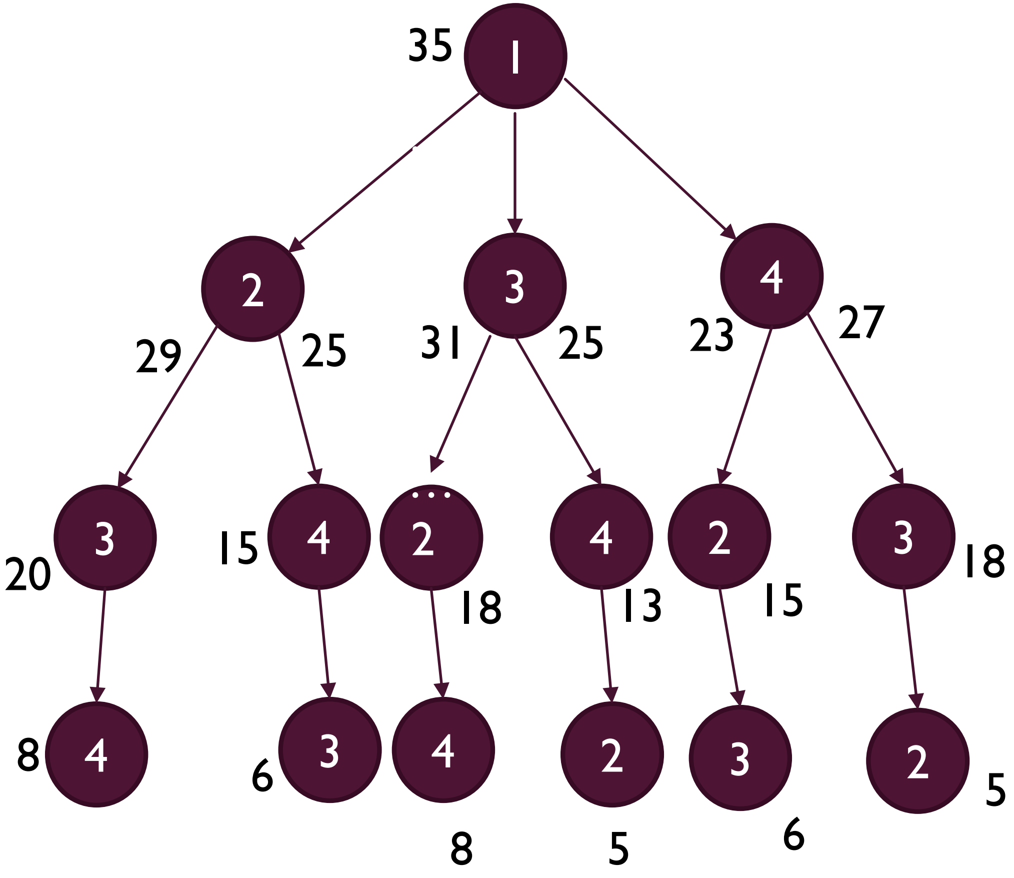


$$g(i,S)=\min\{C_{i,k} + g(k, S-\{k\})\} \quad k\in S$$

$$g(1,\{2,3,4\})=\min\{C_{1,k} + g(k,\{2,3,4\}-\{k\})\}$$

TRAVELING SALESMAN PROBLEM
$g(i,S)=\min\{C_{i,k} + g(k, S-\{k\})\}$ DYNAMIC PROGRAMMING APPROACH

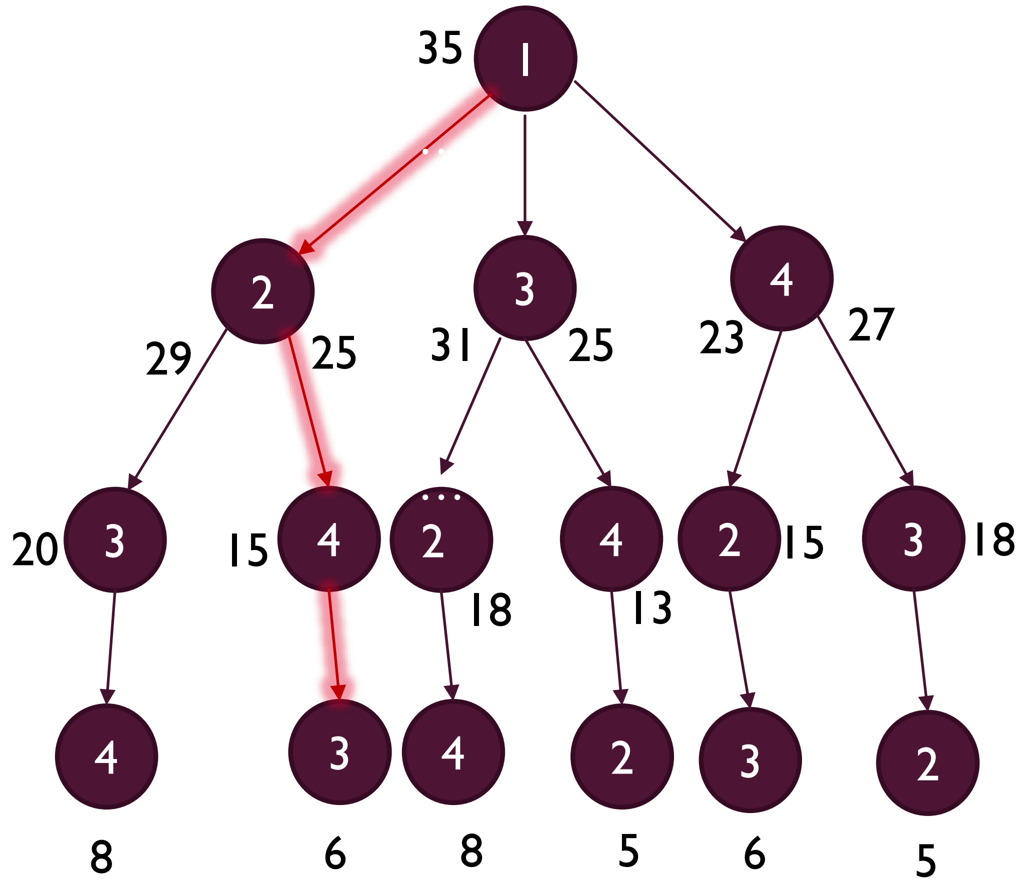
$g(1,\{2,3,4\})=\min\{C_{1,k} + g(k,\{2,3,4\}-\{k\})\}$

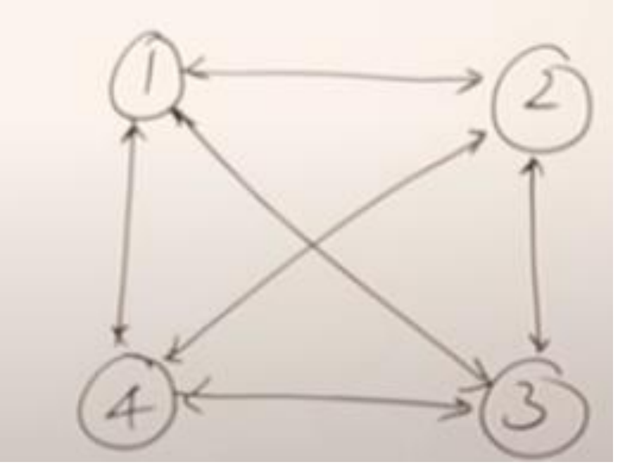


g(2,∅)	5
g(3,∅)	6
g(4,∅)	8
g(2,{3})	15
g(2,{4})	18
g(3,{2})	18
g(3,{4})	20
g(4,{2})	13
g(4,{3})	15
g(2,{3,4})	25
g(3,{2,4})	25
g(4,{2,3})	23

TRAVELING SALESMAN PROBLEM
OUTCOME DYNAMIC PROGRAMMING APPROACH

PATH→ 1→2→4→3



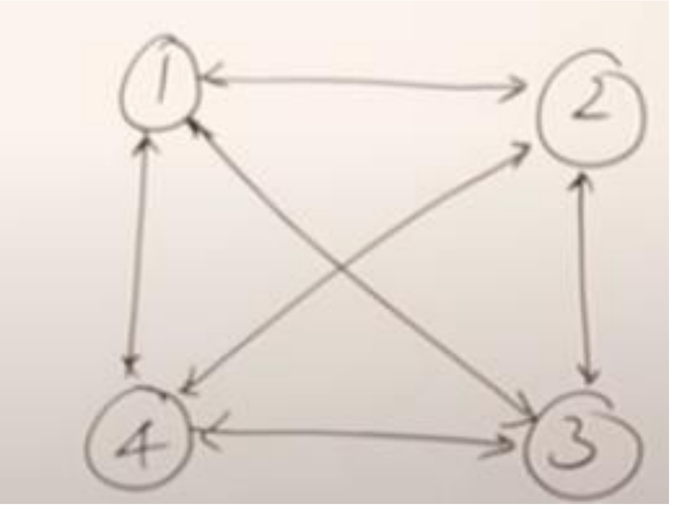


$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix} \end{array}$$

TRAVELING SALESMAN PROBLEM ANALYSIS DYNAMIC PROGRAMMING APPROACH

$$g(i,S)=\min\{C_{i,k} + g(k, S-\{k\})\}$$

- |S|=n (no of nodes)
- There are at most $O(n*2^n)$ sub-problems.
- Each sub-problem takes linear time to solve.
- The total running time is therefore $O(n^2*2^n)$.



TRAVELING SALESMAN PROBLEM

- **Combinatorial optimization problems.**
- **Exponential** in terms of time complexity.
- Requires **exploring all possible permutations** in worst case.

- The **Branch and Bound** Algorithm technique solves these problems relatively quickly.

NEXT CLASS

- Travelling Salesman Problem using Branch & Bound.