Dynamic Programming-Travelling Salesperson Problem

Kumkum Saxena

Travelling Salesman Problem

- Given a complete, weighted graph on n nodes, find the least weight Hamiltonian cycle, a cycle that visits every node once.
- Though this problem is easy enough to explain, it is very difficult to solve.
- Finding all the Hamiltonian cycles of a graph takes exponential time. Therefore, TSP is in the class NP.

If the TSP is so hard, why bother?

The TSP is interesting because it is in the class that is called NP-complete. If the TSP is found to have an algorithm that does not take exponential time, the other problems that are NP-complete will also be solved, and vice-a-versa.

If the TSP is so hard, why bother?

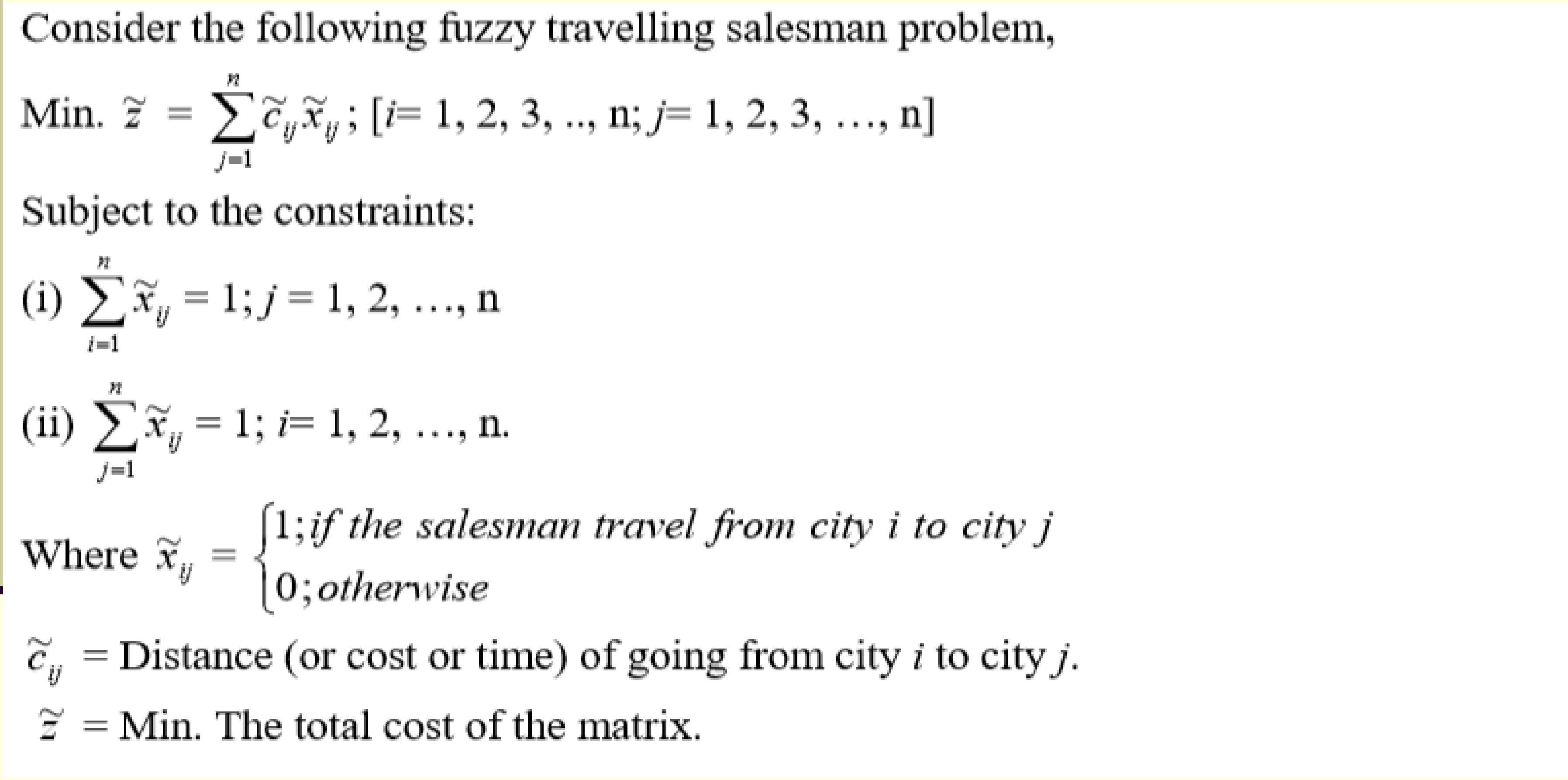
The TSP has many practical applications
-manufacturing
-plane routing
-telephone routing
-networks
-traveling salespeople
-structure of crystals

Problem Statement

- A traveller needs to visit all the cities from a list, where distances between all the cities are known and each city should be visited just once. What is the shortest possible route that he visits each city exactly once and returns to the origin city?

- Travelling salesman problem is the most notorious computational problem. We can use brute-force approach to evaluate every possible tour and select the best one. For **n** number of vertices in a graph, there are **(*n* - 1)!** number of possibilities.
- Instead of brute-force using dynamic programming approach, the solution can be obtained in lesser time, though there is no polynomial time algorithm.
- Let us consider a graph ***G = (V, E)***, where ***V*** is a set of cities and ***E*** is a set of weighted edges. An edge ***e(u, v)*** represents that vertices ***u*** and ***v*** are connected. Distance between vertex ***u*** and ***v*** is ***d(u, v)***, which should be non-negative.

- Suppose we have started at city ***1*** and after visiting some cities now we are in city ***j***. Hence, this is a partial tour. We certainly need to know ***j***, since this will determine which cities are most convenient to visit next. We also need to know all the cities visited so far, so that we don't repeat any of them. Hence, this is an appropriate sub-problem.
- For a subset of cities ***S Є {1, 2, 3, ... , n}*** that includes ***1***, and ***j Є S***, let ***C(S, j)*** be the length of the shortest path visiting each node in **S** exactly once, starting at ***1*** and ending at ***j***.



Consider the following fuzzy travelling salesman problem,

Min. $\tilde{z} = \sum_{j=1}^{n} \tilde{c}_{ij}\tilde{x}_{ij}$; [i= 1, 2, 3, .., n; j= 1, 2, 3, …, n]

Subject to the constraints:

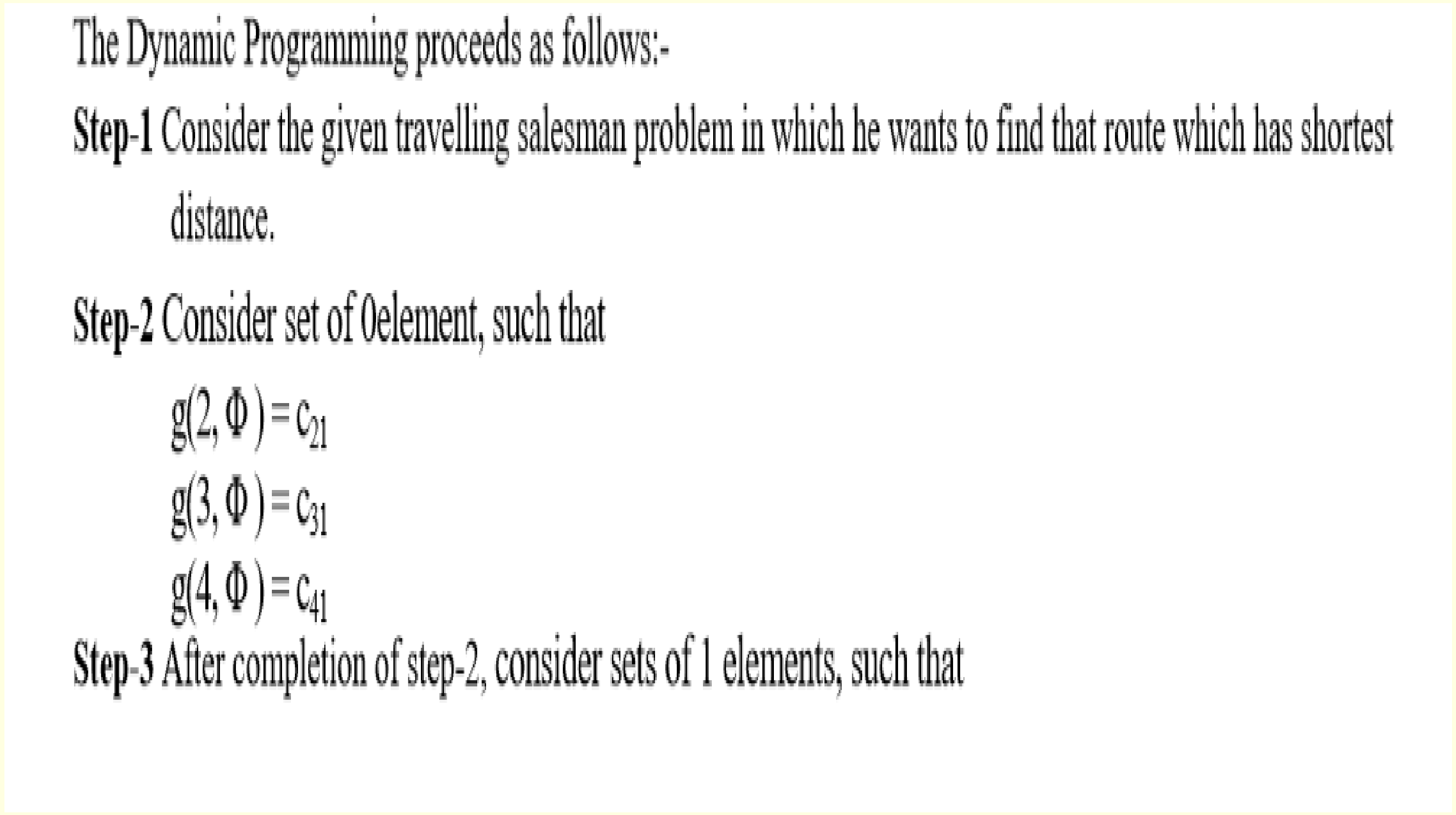
(i) $\sum_{i=1}^{n} \tilde{x}_{ij} = 1; j = 1, 2, \ldots, n$

(ii) $\sum_{j=1}^{n} \tilde{x}_{ij} = 1; i = 1, 2, \ldots, n.$

Where $\tilde{x}_{ij} = \begin{cases} 1; \textit{if the salesman travel from city i to city j} \\ 0; \textit{otherwise} \end{cases}$

$\tilde{c}_{ij}$ = Distance (or cost or time) of going from city i to city j.

$\tilde{z}$ = Min. The total cost of the matrix.



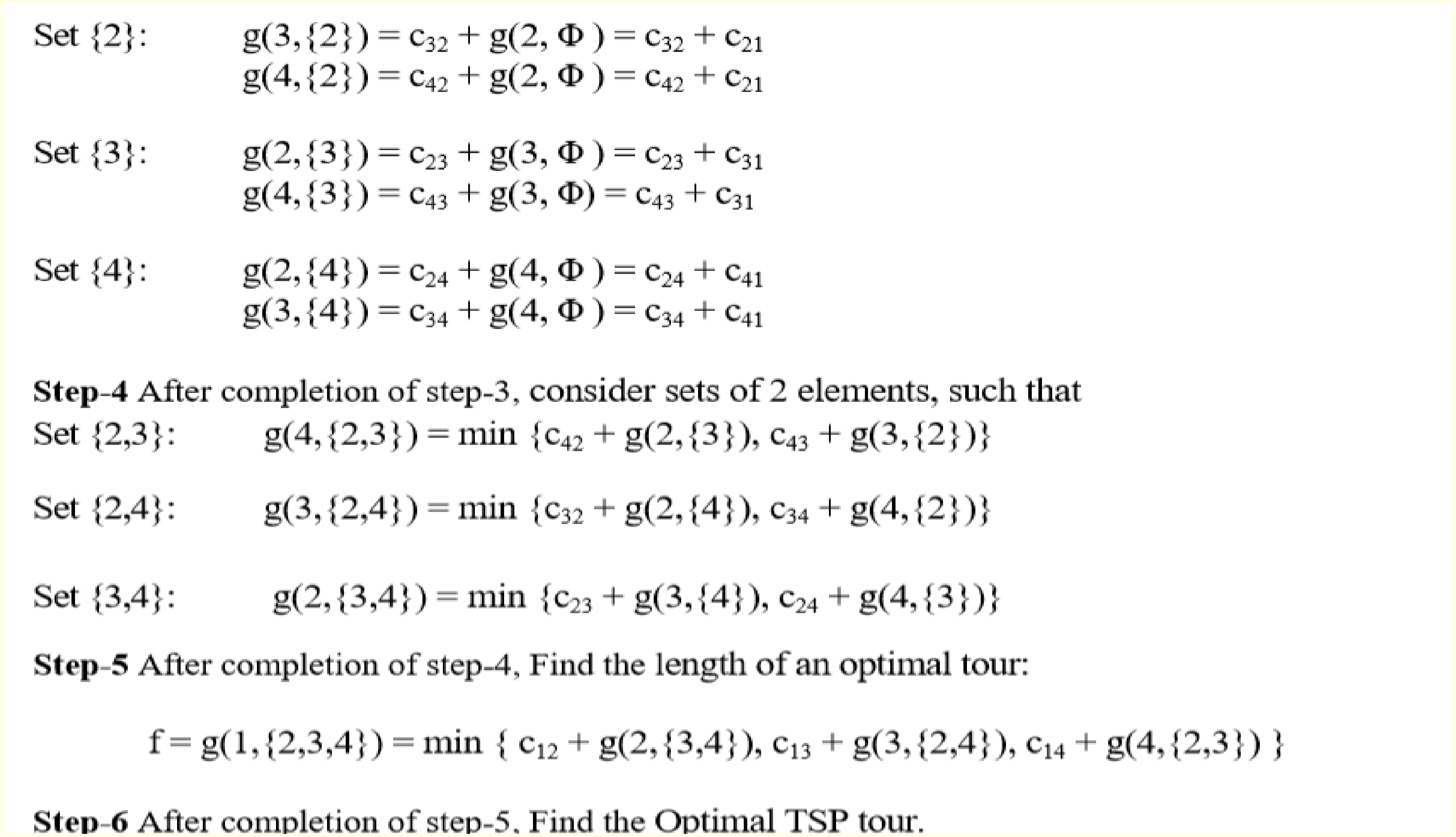
The Dynamic Programming proceeds as follows:-

Step-1 Consider the given travelling salesman problem in which he wants to find that route which has shortest distance.

Step-2 Consider set of 0element, such that

$$g(2, \Phi) = c_{21}$$
$$g(3, \Phi) = c_{31}$$
$$g(4, \Phi) = c_{41}$$

Step-3 After completion of step-2, consider sets of 1 elements, such that



Set {2}: $g(3,\{2\}) = c_{32} + g(2, \Phi) = c_{32} + c_{21}$

$g(4,\{2\}) = c_{42} + g(2, \Phi) = c_{42} + c_{21}$

Set {3}: $g(2,\{3\}) = c_{23} + g(3, \Phi) = c_{23} + c_{31}$

$g(4,\{3\}) = c_{43} + g(3, \Phi) = c_{43} + c_{31}$

Set {4}: $g(2,\{4\}) = c_{24} + g(4, \Phi) = c_{24} + c_{41}$

$g(3,\{4\}) = c_{34} + g(4, \Phi) = c_{34} + c_{41}$

Step-4 After completion of step-3, consider sets of 2 elements, such that

Set {2,3}: $g(4,\{2,3\}) = \min \{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\}$

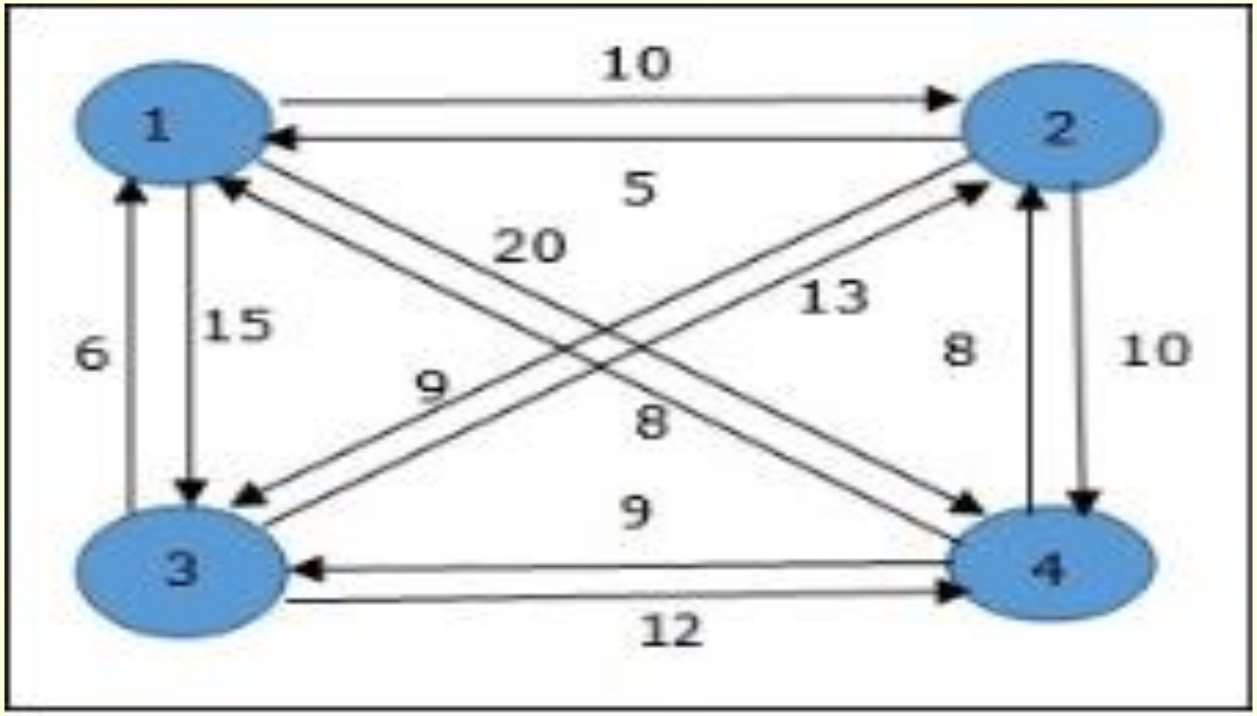
Set {2,4}: $g(3,\{2,4\}) = \min \{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\}$

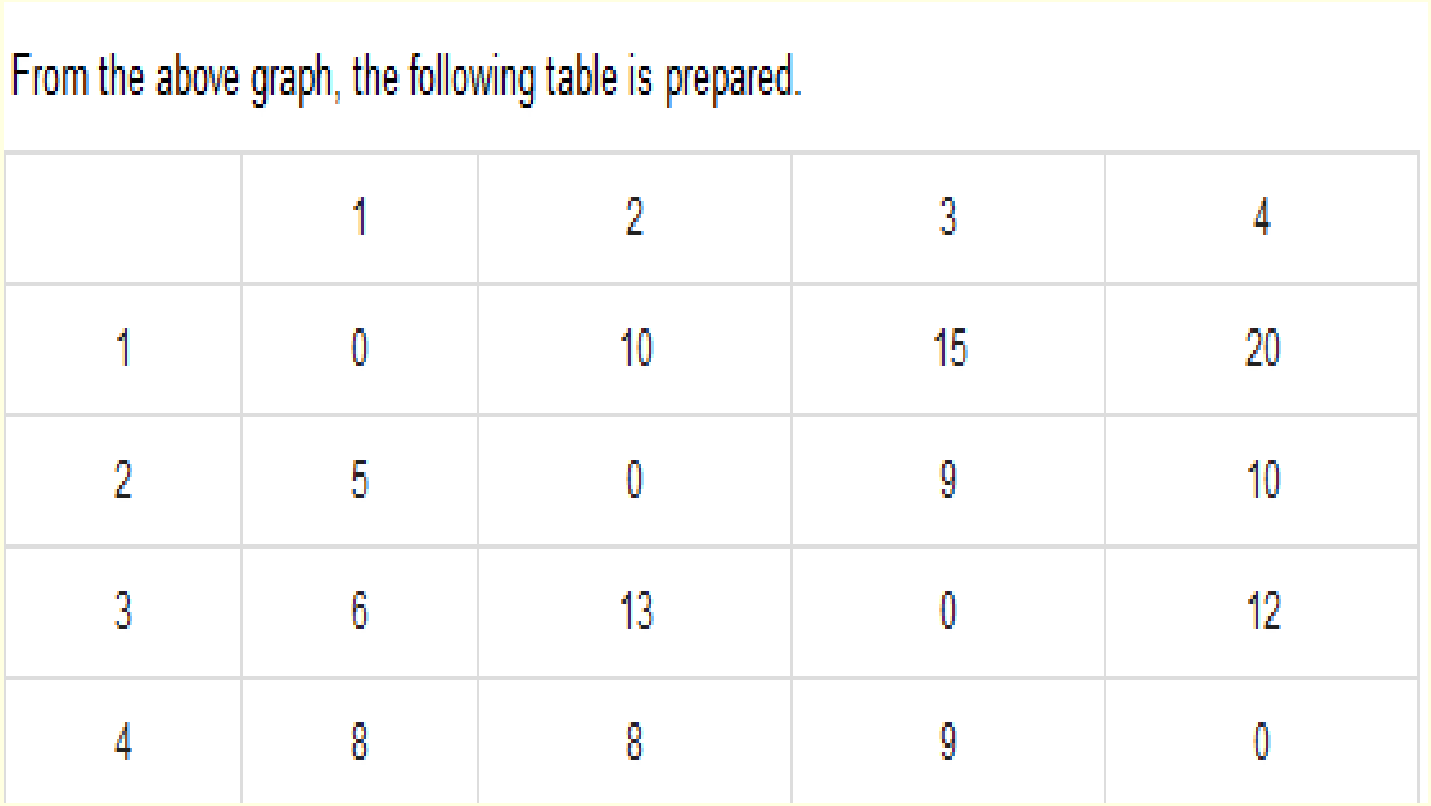
Set {3,4}: $g(2,\{3,4\}) = \min \{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\}$

Step-5 After completion of step-4, Find the length of an optimal tour:

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

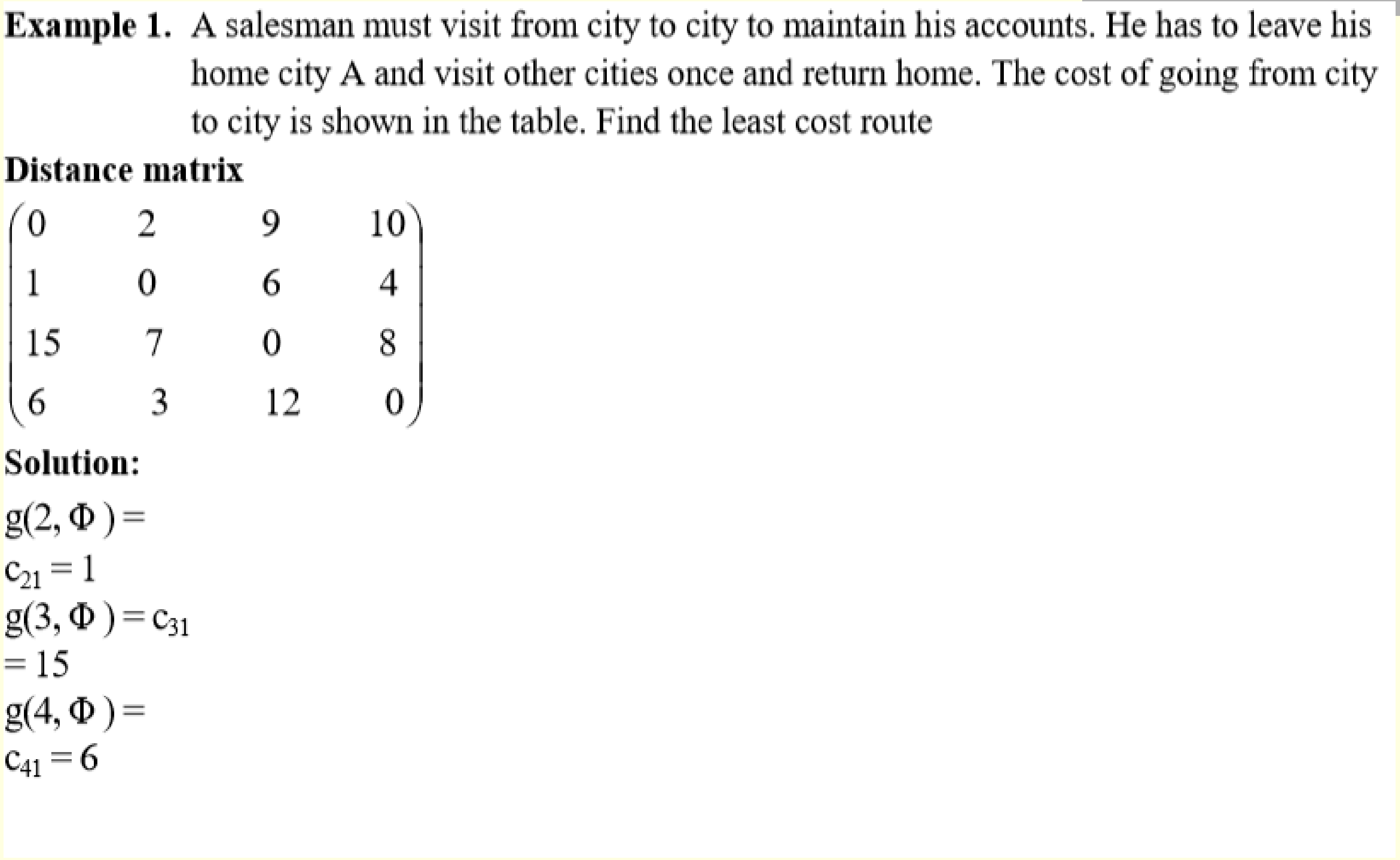
Step-6 After completion of step-5, Find the Optimal TSP tour.


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



From the above graph, the following table is prepared.

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



Example 1. A salesman must visit from city to city to maintain his accounts. He has to leave his home city A and visit other cities once and return home. The cost of going from city to city is shown in the table. Find the least cost route

Distance matrix

$$\begin{pmatrix} 0 & 2 & 9 & 10 \\ 1 & 0 & 6 & 4 \\ 15 & 7 & 0 & 8 \\ 6 & 3 & 12 & 0 \end{pmatrix}$$

Solution:

$g(2, \Phi) =$

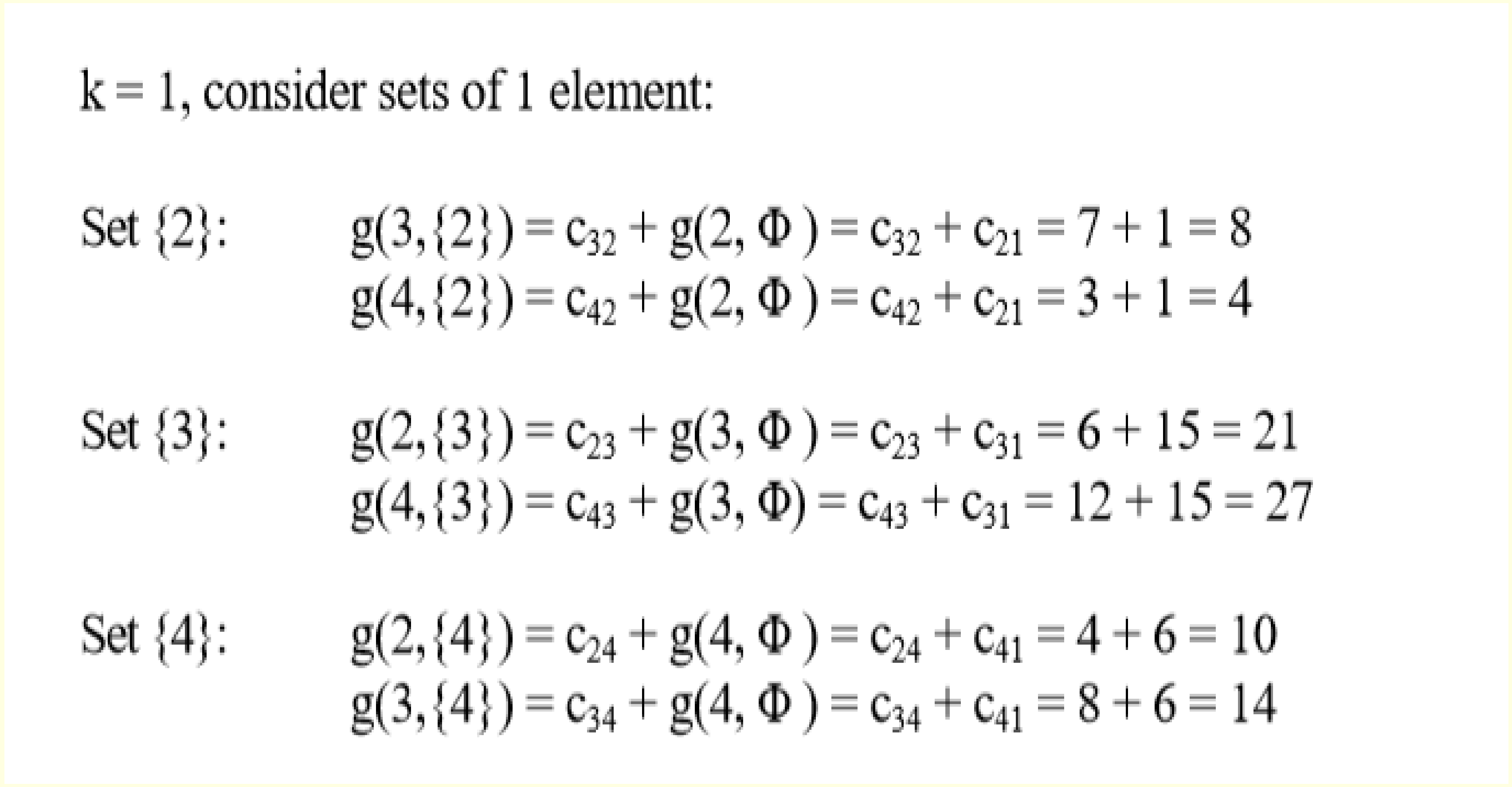
$c_{21} = 1$

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

$= 15$

$g(4, \Phi) =$

$c_{41} = 6$



k = 1, consider sets of 1 element:

Set {2}: $g(3,\{2\}) = c_{32} + g(2, \Phi) = c_{32} + c_{21} = 7 + 1 = 8$

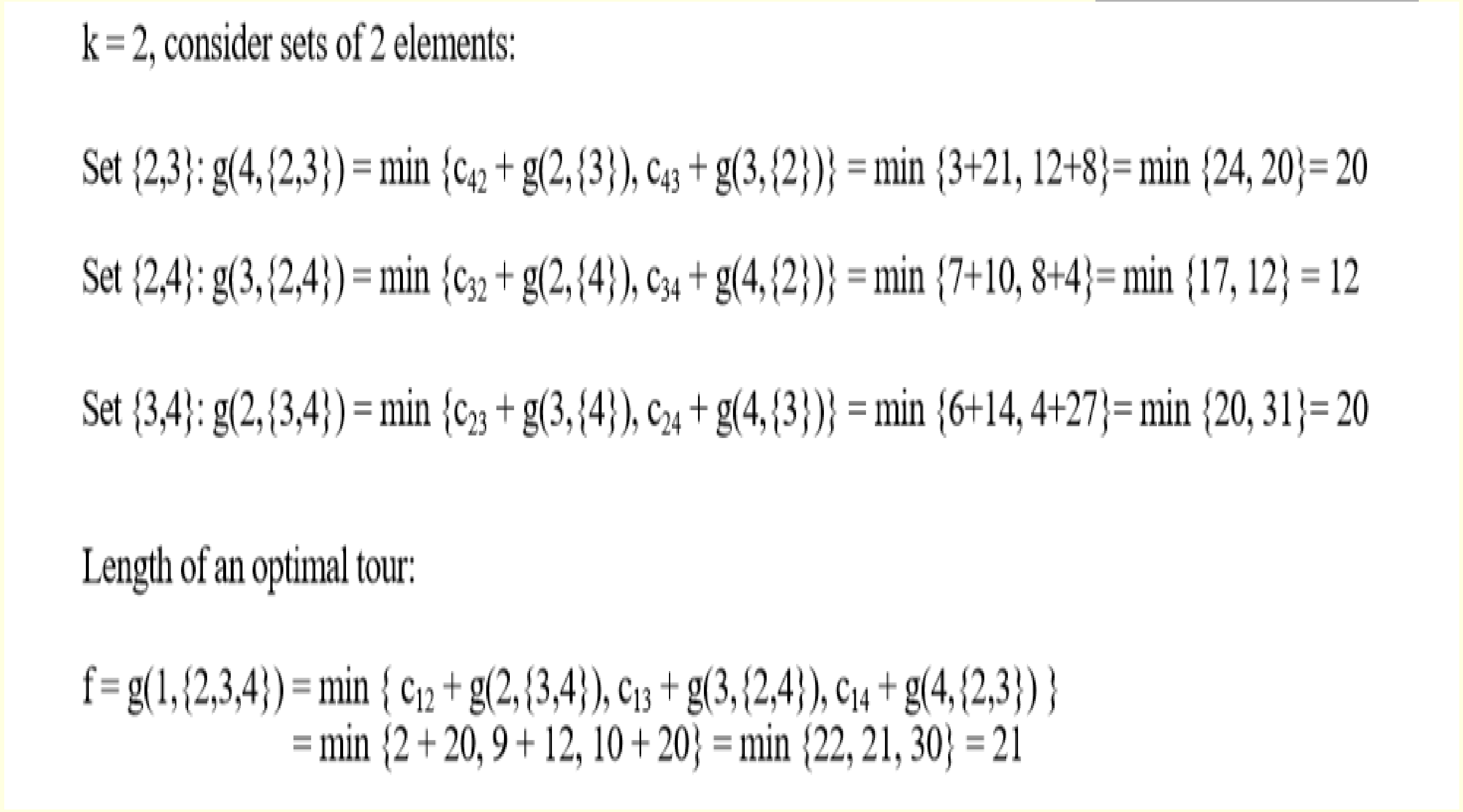
$g(4,\{2\}) = c_{42} + g(2, \Phi) = c_{42} + c_{21} = 3 + 1 = 4$

Set {3}: $g(2,\{3\}) = c_{23} + g(3, \Phi) = c_{23} + c_{31} = 6 + 15 = 21$

$g(4,\{3\}) = c_{43} + g(3, \Phi) = c_{43} + c_{31} = 12 + 15 = 27$

Set {4}: $g(2,\{4\}) = c_{24} + g(4, \Phi) = c_{24} + c_{41} = 4 + 6 = 10$

$g(3,\{4\}) = c_{34} + g(4, \Phi) = c_{34} + c_{41} = 8 + 6 = 14$



k = 2, consider sets of 2 elements:

Set {2,3}: $g(4,\{2,3\}) = \min\{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\} = \min\{3+21, 12+8\} = \min\{24, 20\} = 20$

Set {2,4}: $g(3,\{2,4\}) = \min\{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\} = \min\{7+10, 8+4\} = \min\{17, 12\} = 12$

Set {3,4}: $g(2,\{3,4\}) = \min\{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\} = \min\{6+14, 4+27\} = \min\{20, 31\} = 20$

Length of an optimal tour:

$$f = g(1,\{2,3,4\}) = \min\{ c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\}) \}$$
$$= \min\{2 + 20, 9 + 12, 10 + 20\} = \min\{22, 21, 30\} = 21$$

Successor of node 1: $g(2, \Phi) = c_{21} = 1$

Successor of node 2: $g(4, \{2\}) = 4$

Successor of node 4: $g(3, \{2,4\}) = 12$

Successor of node 3: $g(1, \{2,3,4\}) = 21$

Optimal TSP tour: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

Travelling Salesman Problem

for each subset S of the cities with
$|S| \geq 2$ and each $u,v \in S$

K[S,u,v] the length of the shortest path that
- starts at u
- ends at v
- visits all cities in S

How large is K ?

Travelling Salesman Problem

for each subset S of the cities with
$|S| \geq 2$ and each $u,v \in S$

K[S,u,v] the length of the shortest path that
- starts at u
- ends at v
- visits all cities in S

How large is K ?

$\approx 2^n n^2$

Travelling Salesman Problem

K[S,u,v]

some vertex w $\in$ S – {u,v}
must be visited first

d(u,w) = we get to w
K[S-u,w,v] = we need to get from w to v
and visit all vertices in S-u

Travelling Salesman Problem

K[S,u,v] the length of the shortest path that
- starts at u
- ends at v
- visits all cities in S

if S={u,v} then K[S,u,v]=d(u,v)

if |S|>2 then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

Travelling Salesman Problem

if S={u,v} then K[S,u,v]=d(u,v)

if |S|>2 then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

Running time = ?

$K \approx 2^n n^2$

Travelling Salesman Problem

if S={u,v} then K[S,u,v]=d(u,v)

if |S|>2 then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

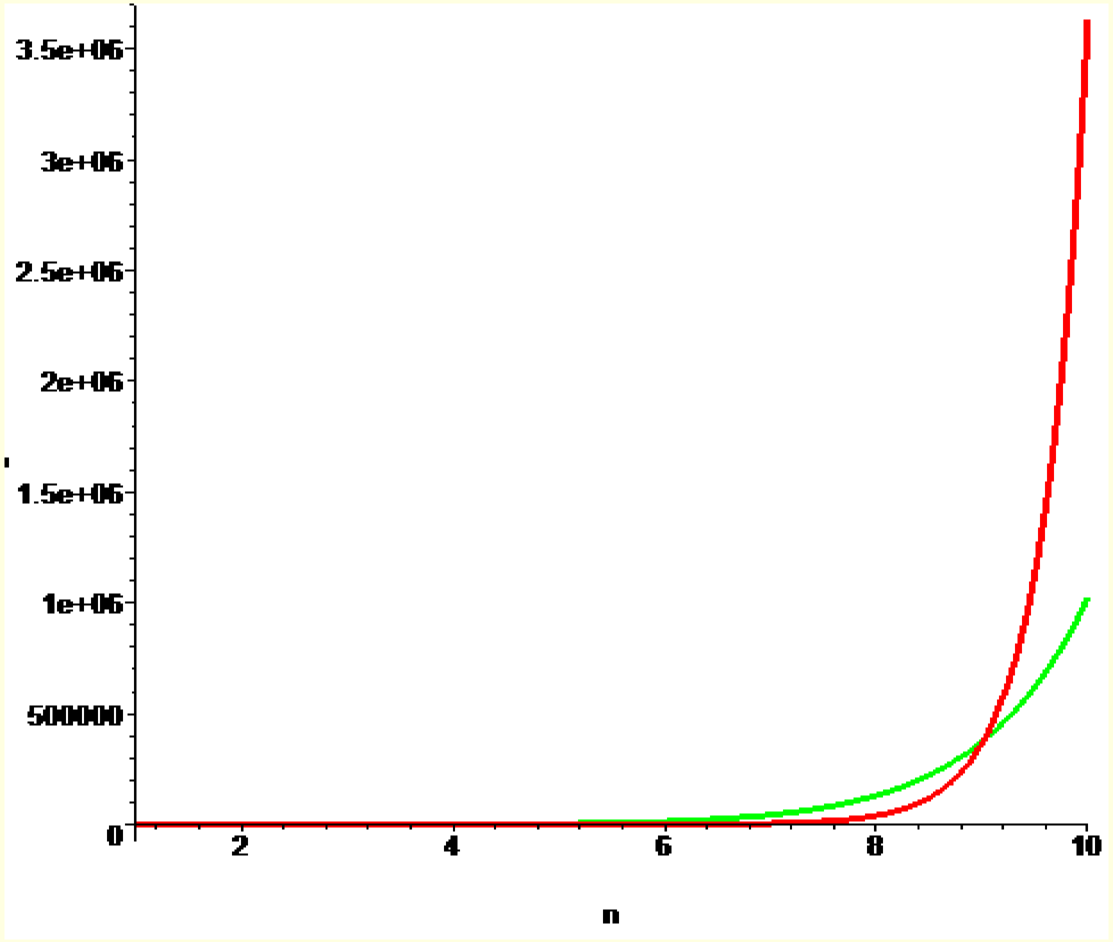
Running time = $O(n^3 2^n)$

$K \approx 2^n n^2$

Travelling Salesman Problem

dynamic programming = $O(n^3 2^n)$

brute force = $O(n!)$



Analysis

- There are at the most $(2^n.n)$ sub-problems and each one takes linear time to solve. Therefore, the total running time is $O(2^n.n^2)$.