SNS COLLEGE OF ENGINEERING

Coimbatore-107

COURSE NAME: ANALYSIS OF ALGORITHM

II YEAR/ IV SEMESTER

UNIT – V

BRANCH& BOUND ALGORITHM

Topic

Traveling Salesman Problem

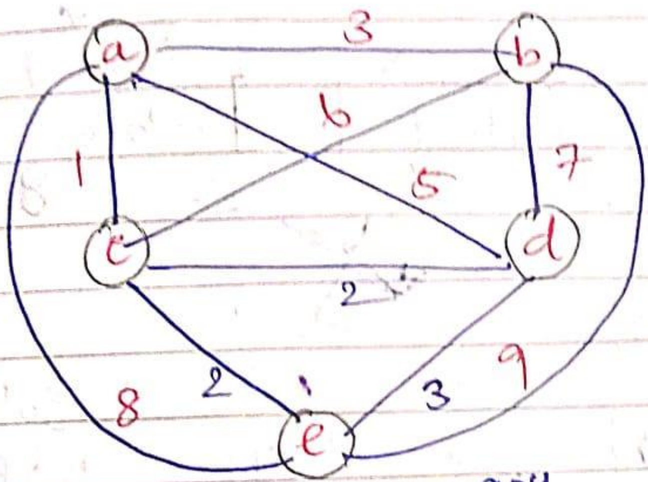
UNIT 5

Travelling Salesman Problem

classmate Date Page

Example:

Logic → Salesman traverse all the cities & come back to same city where he has started.



Step 1: Start from any vertex; check the neighbouring node it traverse. Select the two cities with minimum cost.

a → ab + ac = 3 + 1 = 4

b → ba + bc = 3 + 6 = 9

c → ca + ce = 1 + 2 = 3

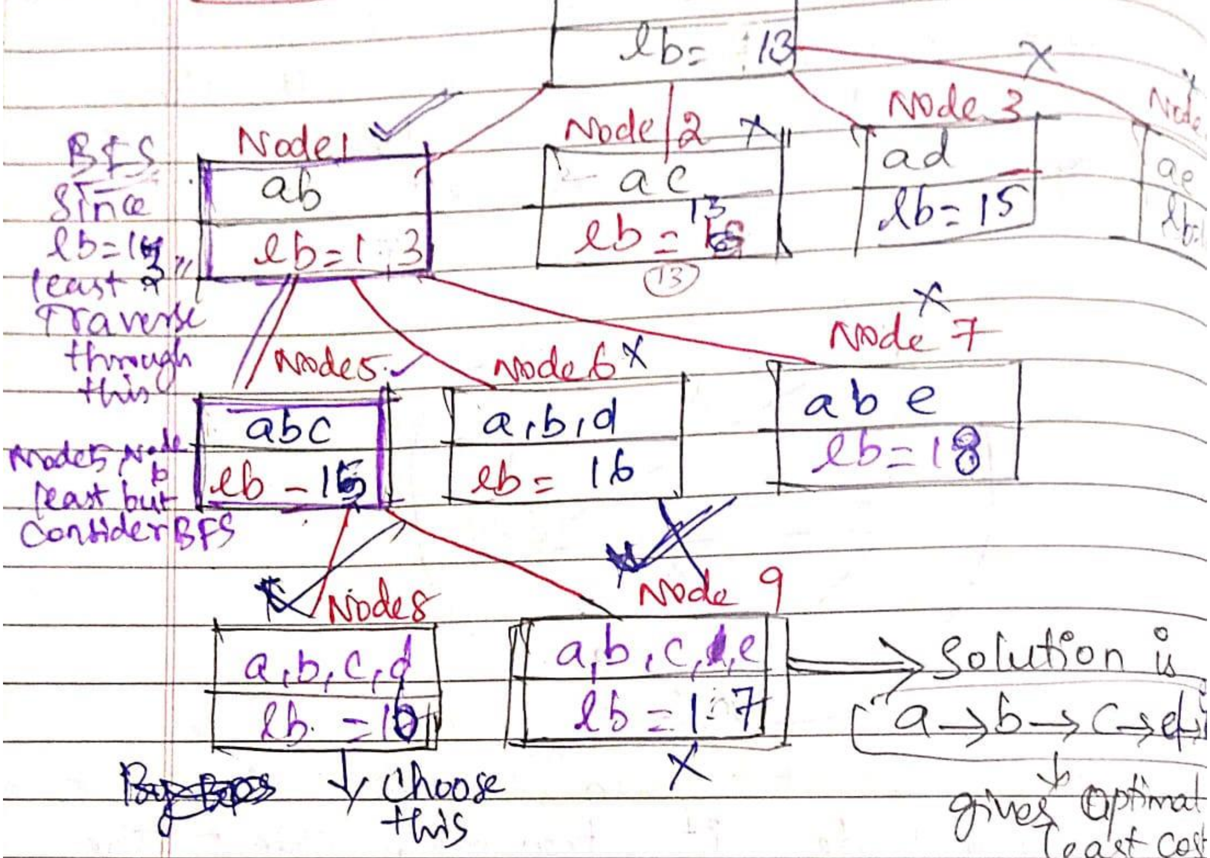
d → dc + de = 2 + 3 = 5

e → ec + ed = 2 + 3 = 5

26

$lb = \left[\frac{s}{2}\right] = \frac{26}{2} = 13$

Step 2:



Step 3: Node 1 : "ab"

$a \rightarrow ab + ac$ (Next least) $= 3 + 1 = 4$

$b \rightarrow ab + bc$ (Next least) $= 3 + 6 = 9$

$c \rightarrow 3$ (Previous Node value)

$d \rightarrow 5$ (Previous Node value)

$e \rightarrow 5$ (Previous Node value)

26

$$lb = \left\lceil \frac{26}{2} \right\rceil = 13$$

classmate

Date

Page

Step 4 : Node 2 : a, c (next least from a)

a → ac + ab = 1 + 3 = 4

b → 9

c → ac + ce = 1 + 2 = 3

d → 5

e → 5

26

$lb = \left\lceil \frac{26}{2} \right\rceil = 13.$

Step 5 : Node 3 : a, d (least from a)

a → ad + ac = 5 + 1 = 6

b → 9

c → 3

d → ad + de = 5 + 2 = 7 (least from d)

e → 5

30

$lb = \frac{30}{2} = 15.$

Step 6 : Node 4 : a, e (least from a)

a → ae + ac = 8 + 1 = 9

b → 9

c → 3

d → 5

e → ae + ec = 8 + 2 = 10 (least from e)

36

$lb = \frac{36}{2} = 18$

Step 7: Node 5: "a, b, c" = ab, bc

Since a [illegible] update ab from a → ab + ac = 3+1 = 4

Since b [illegible] update the → b → ab + bc = 3+6 = 9

Since c in bc update as first value → c → bc + ac = 6+1 = 7

For second value take least from node 0 → d → 7 = 7

e → 5 = 3

30

For remaining just update from Node 0.

$lb = \frac{30}{2} = 15$

$lb = 15$

Step 8: Node 6: "a, b, d" ⇒ ab, bd

a → ab + ac = 3+1 = 4

b → ab + bd = 3+7 = 10

c → 3 = 3

d → bd + dc = 7+2 = 9

e → 5 = 5

31

$lb = \lceil \frac{31}{2} \rceil = \lceil 15.5 \rceil = 16.$

Step 9: Node 7: "a, b, e" ⇒ ab, be

a → ab + ac = 3+1 = 4

b → ab + be = 3+9 = 12

c → 3 = 3

d → 5 = 5

e → be + ec = 9+2 = 11

35

$lb = \left\lceil \frac{35}{2} \right\rceil = \lceil 17.5 \rceil = 18$

lb = 18

Node 8:

Step 9: a, b, c, d => ab, bc, cd

a → ab + ac = 3 + 1 = 4

b → ab + bc = 3 + 6 = 9

c → bc + cd = 6 + 2 = 8

minimum cost → d → cd + de = 2 + 3 = 5

e → 5 = 5

31

$lb = \left\lceil \frac{31}{2} \right\rceil = 15.5 = 16$

Node 9:

Step 10: a, b, c, e => ab, bc, ce

a → ab + ac = 3 + 1 = 4

b → ab + bc = 3 + 6 = 9

c → bc + ce = 6 + 2 = 8

d → 7 = 7 = 7

e → ce + de = 2 + 3 = 5

33

$lb = \left\lceil \frac{33}{2} \right\rceil = 16.5 = 17$

Final Solution = a – b – c – d → e → a

This have minimum cost

(Branch & Bound)

(Travelling Salesman)

Algorithm:

bound (currpath, n, graph, pos) // calculate bo

if (pos==n) // Function to perform Branch & Bound

{ //All cities visited, complete tour & Reset to 0]

if (graph(currpath [pos-1] [currpath[0]] != 0)

{ currcost += graph [currpath [pos-1]] [curr

bestcost = min (bestcost, currcost)

} }

for (i=0; i<n; i++) //Loop all city

{ if (not visited [i] && graph [currpath (pos-

{ visited [i] = true

currpath [pos] = i // calculate bound for

boundcost = currcost + graph [currpath [pos-1]]

Bound (currpath, n, graph, pos)

if (boundcost < bestcost)

{ Branch& Bound (currpath, visited, n, graph

pos+1, currcost + graph [curr

}

}

visited [i] = false }}

Time complexity:

Without pruning:

O(n!): Since it explores all possibilities of the cities to find optimal solu

with pruning: Not Better than O(n)!

Space complexity:

O(n) → depending no. of cities