30/4/21 Prashanth. S

1) Provide a solution to the travelling salesman problem using Approximation Algorithms. Takes at 8 cities and draw the graph. Explain your answer using your graph

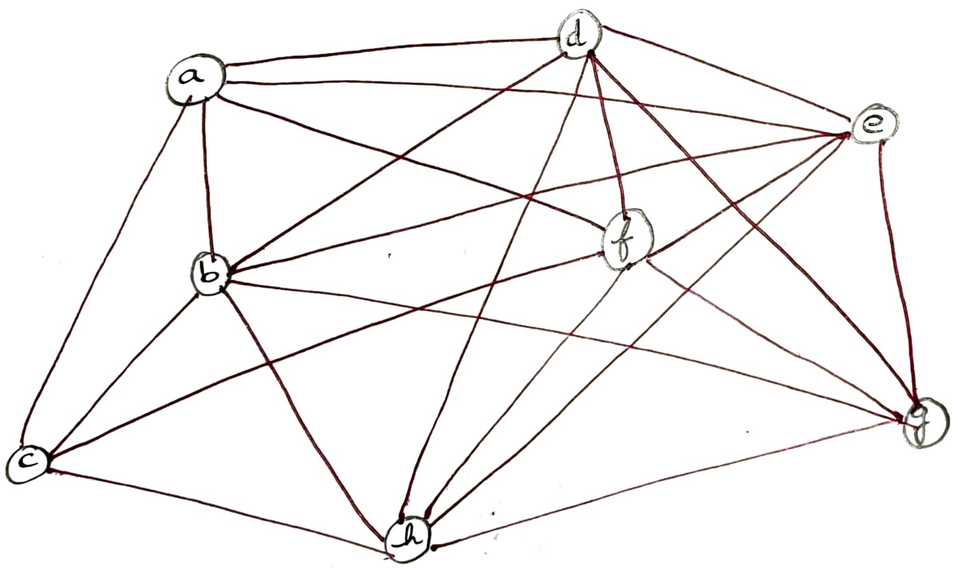
APPROACH

Step - 1 : Compute a minimum spanning tree

Step - 2 : Perform preorder walk i.e) crawl around each node following the edges and finally return back to the first vertex

Step - 3 : Then edges are joined in this order to obtain approximated tour walk

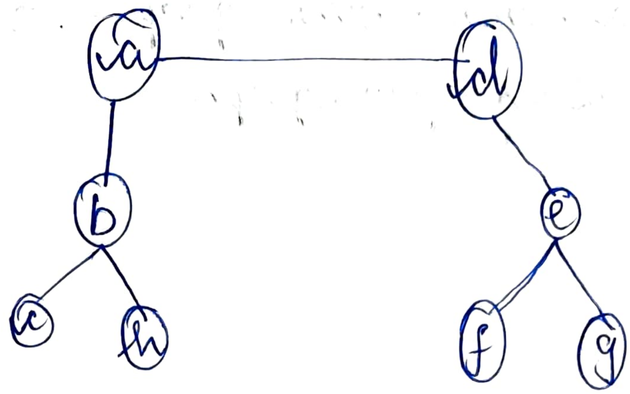
EXAMPLE



In this example all nodes (cities) are connected to all other nodes. Totally 8 cities

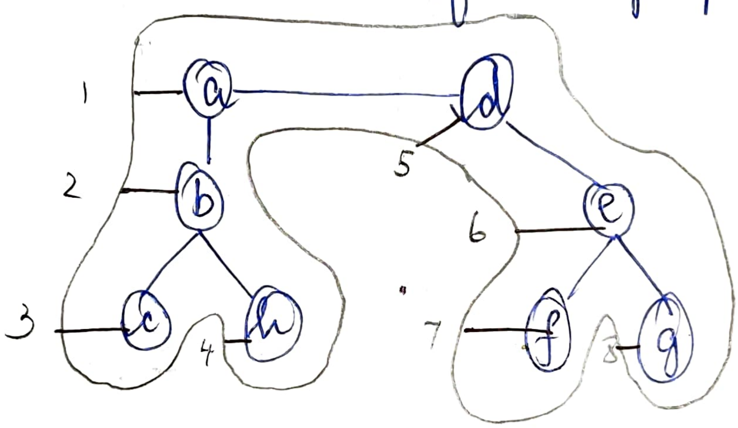
Step - 1

Compute minimum spanning tree



Step - 2

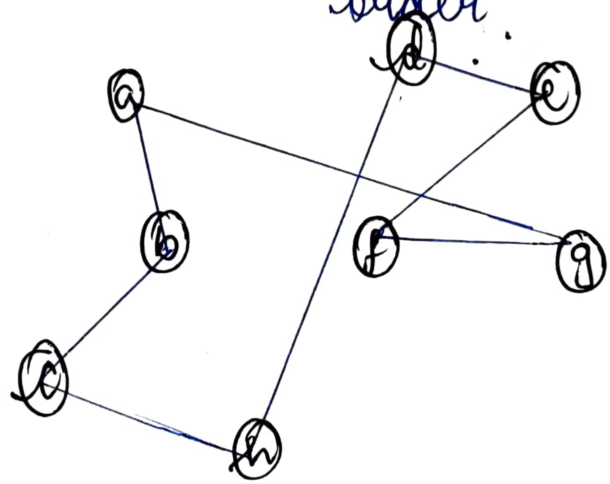
Preorder walk of the graph



Preorder traversal $\Rightarrow a \to b \to c \to h \to d \to e \to f \to g$

Step - 3

TSP solution: join edges in preorder traversal order.



How would you find the shortest path to the goal node from a start node using A* algorithm. Explain it using search graph

Solution

i) Start with OPEN containing only initial node
Set the node's G value to 0,
h' value to whatever it is,
f' value to h' + 0 or h'
Set closed to empty list

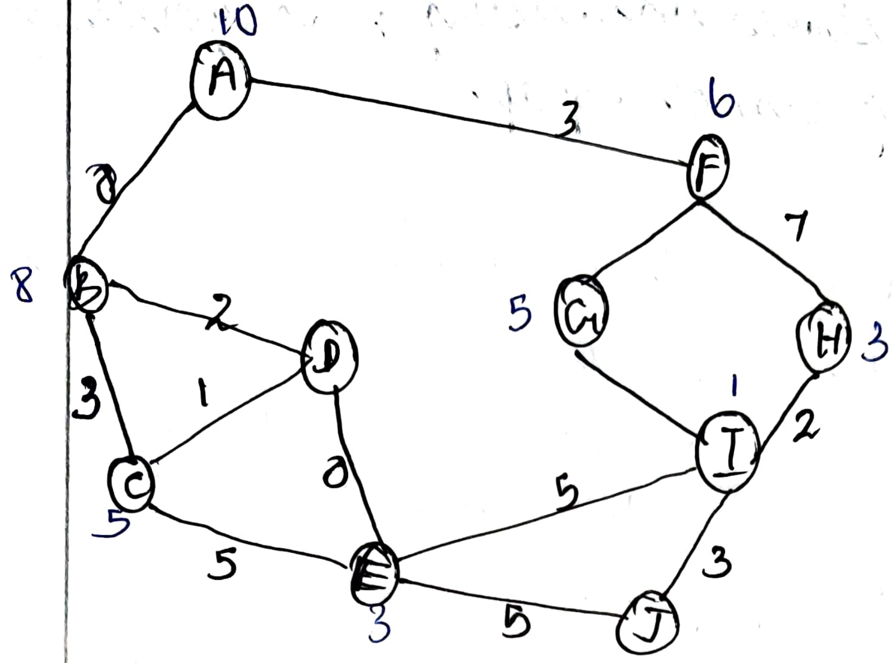
ii) Until a goal node is found, repeat the following procedure. If there are no nodes on OPEN, report failure. Otherwise, pick the node on OPEN with the lowest f' value. Call it BESTNODE. Remove it from OPEN. Place it in CLOSED. See if BESTNODE is a goal state. If so exit and report a solution otherwise, generate the successors of BESTNODE but do not set the BESTNODE to point to them yet.

iii) For each successor, do the following

a) Set SUCCESSOR to point to BESTNODE. These backwards links will make it possible to recover the path once a solution is found

b) compute g(SUCCESSOR) = g(BESTNODE to SUCCESSOR

c) If SUCCESSOR was not already on either OPEN or CLOSED, then put it on OPEN and add it to the list of BESTNODE's successors.
compute f'(SUCCESSOR) = g(SUCCESSOR) + h'(SUCCESSOR)



Step - I

- GOAL = J , START NODE = A
- OPEN = {A}
- g(A) = 0
- h(A) = 0+10 = 10
- f(A) = g(A) + h(A) = 0+10 = 10
- CLOSED = { }
- BESTNODE = {A}
- OPEN = { }
- CLOSED = {A}
- SUCCESSORS of A = {B, F}

Step - 2

- For successor B
  - A ← B
  - g(B) = 0+6 = 6
  - open = {B}
  - f(B) = 6+8 = 14
- For successor

Step - 2

* For successor B

- A ← B
- g(B) = 0+6 = 6
- open = {B}
- f(B) = 6+8 = 14

* For successor F

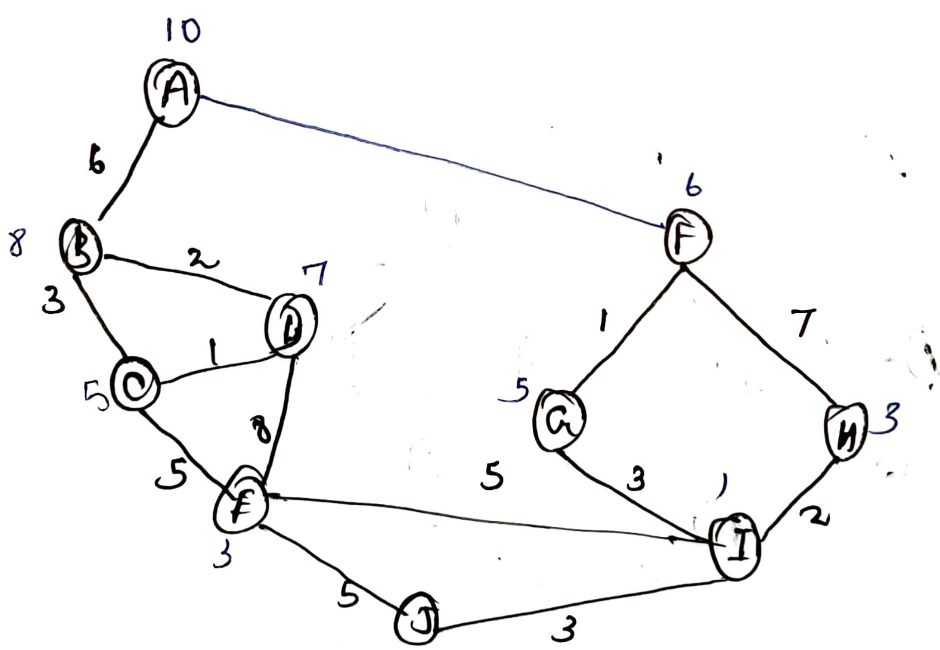
- A ← F
- g(F) = 0+3 = 3
- open = {B, F}
- f(F) = 3+6 = 9

* Bestnode = {F}

Open = {B}

closed = {A, F}

successor of F = {G, H}



Step - 3

* For successor G

  - A ← F ← G
  - g(G) = 4
  - open = {B, G}
  - f(G) = 4 + 5 = 9

* For successor H

  - A ← F ← H

  g(H) = 10

  open = {B, G, H}

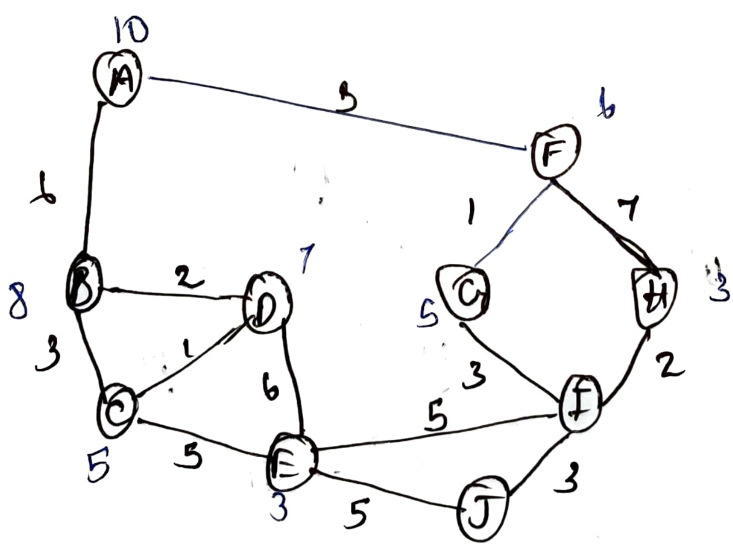
  f(H) = 10 + 3 = 13

* Best node = {G}

  open = {B, G}

  closed = {A, F, G}

  Successor of G = {I}

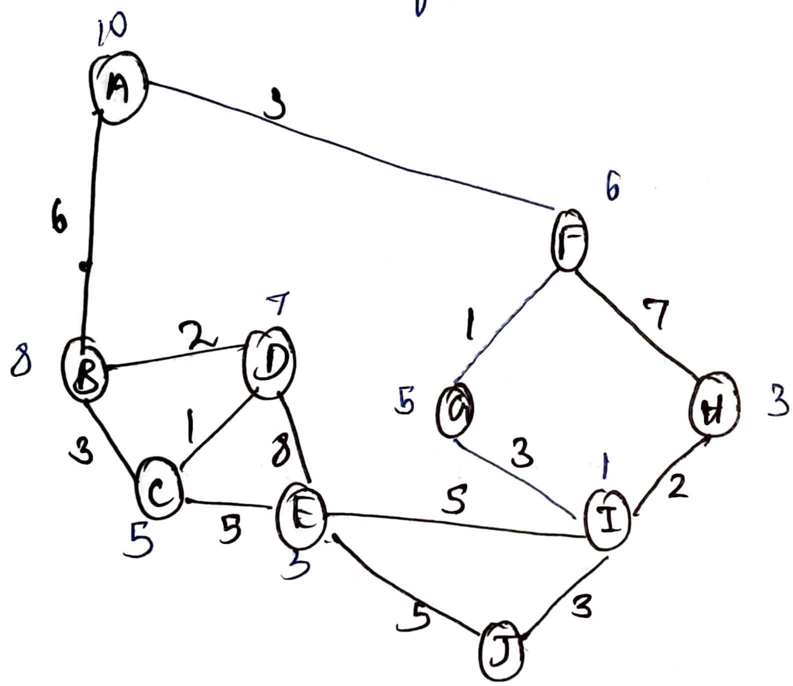


Step - 4

For successor I

- A ← F ← G ← I
- g(I) = 7
- open = {B, H, I}
- f(I) = 7 + 1 = 8

Best node = {I}
open = {B, H}
closed = {A, F, G, I}
Successors of I = {E, J}



Step - 5

- For successor E
  - A ← F ← G ← I ← E
  - g(E) = 7+5 = 12
  - Open = { B, H, E }
  - f(E) = 12+3 = 15

- For successor J
  - A ← F ← G ← I ← J
  - g(J) = 7+3 = 10
  - open = { B, H, J }
  - f(J) = 10+0 = 10

- Best node = { J }

open = { B, H }

closed = { A, F, G, I, J }

Goal node found

The set ~~closed~~ CLOSED contains the shortest path

A ← F ← G ← I ← J