Logical Task - 2: Part B

Given,
The bidirectional graph has nodes and distances between them.

As:

[['a', 'b', 7],
['b', 'c', 3],
['d', 'e', 4],
['d', 'e', 1],
['f', 'c', 6],
['g', 'e', 3],
['a', 'd', 5],
['h', 'g', 8],
['i', 'h', 9],

['c', 'h', 14]]

ⓐ Draw the graph in digital or hand written form (3 marks)

=)

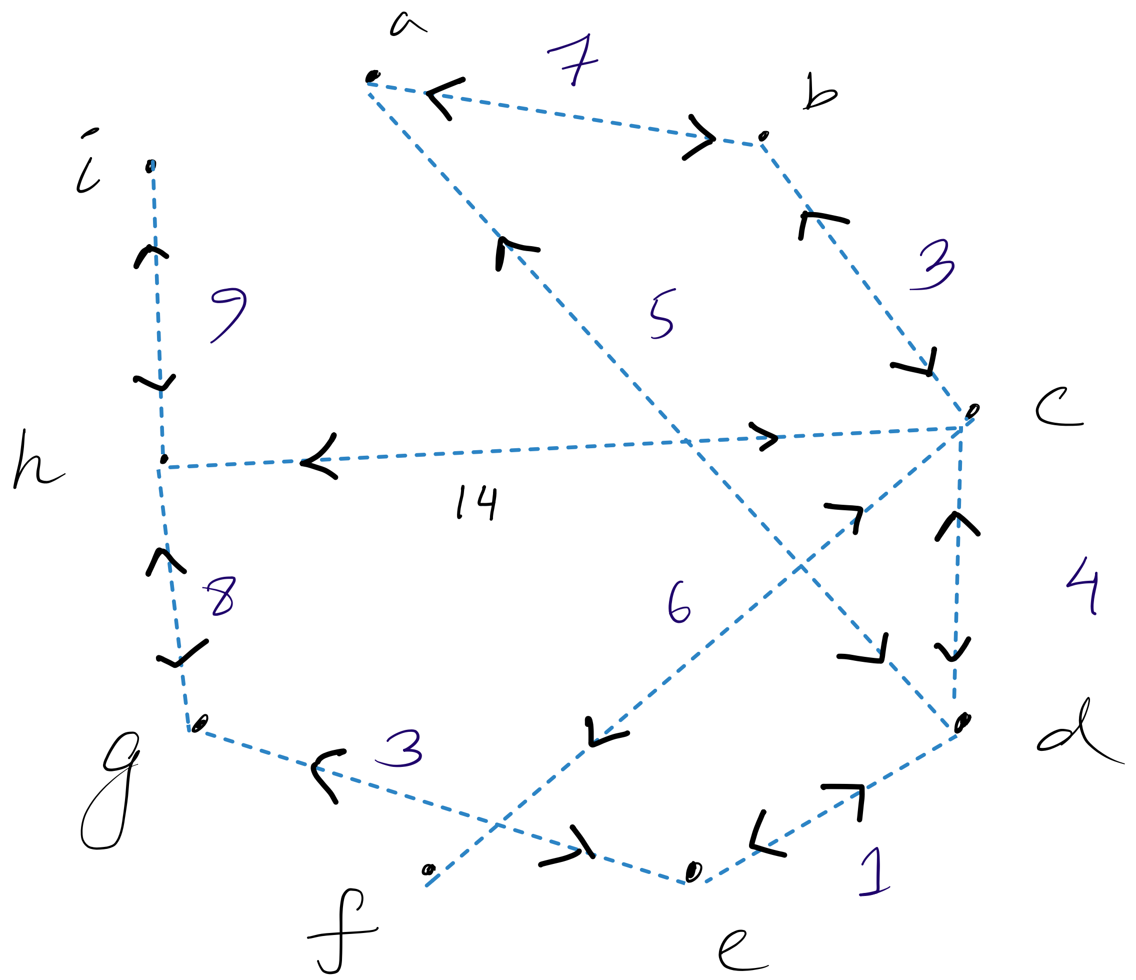


Fig: Bi directional graph with corresponding nodes and distances.

Two directed arrows have been used on each edge as it is a bi-directional graph.

B. Shortest path required from 'a' to 'i' nodes: –

⇒

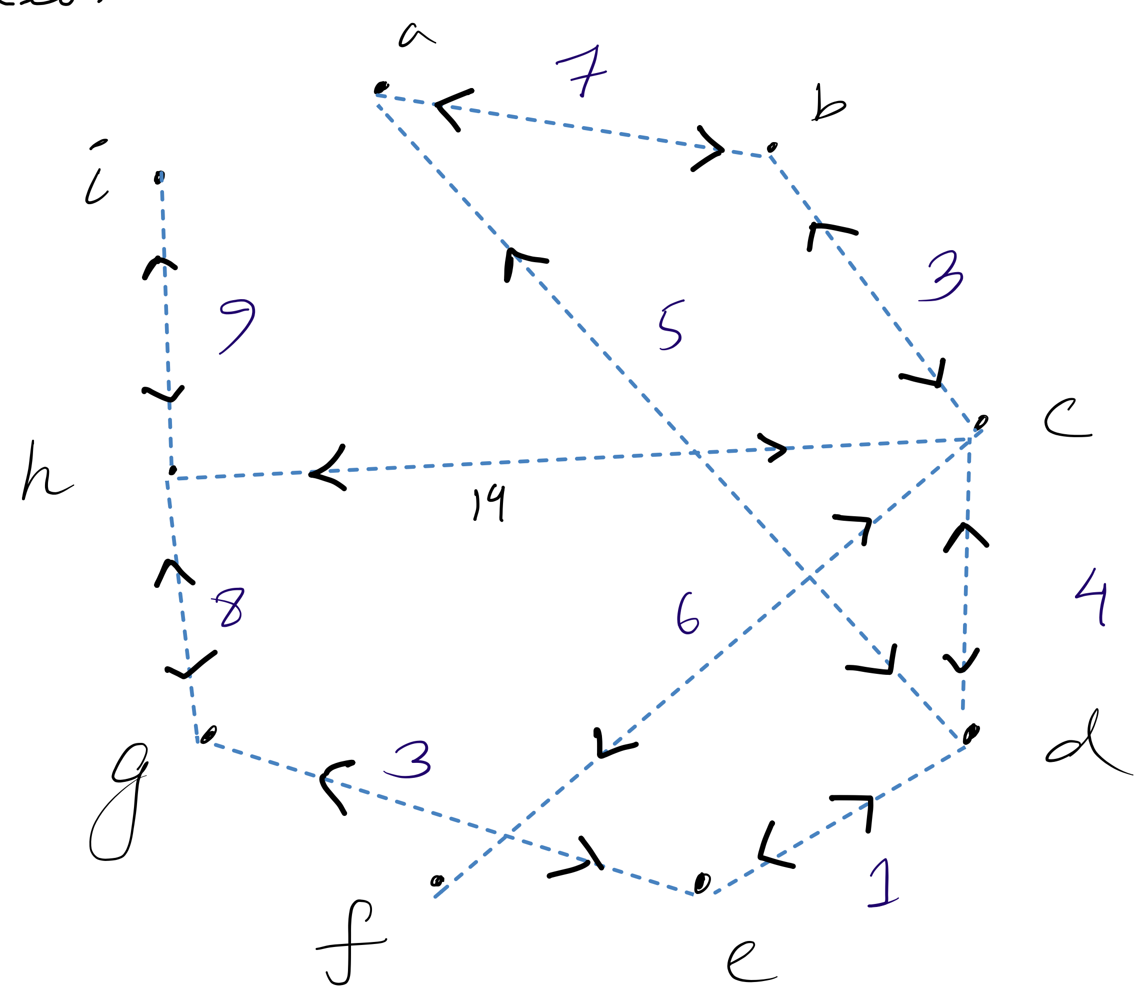


Fig: Graph from before

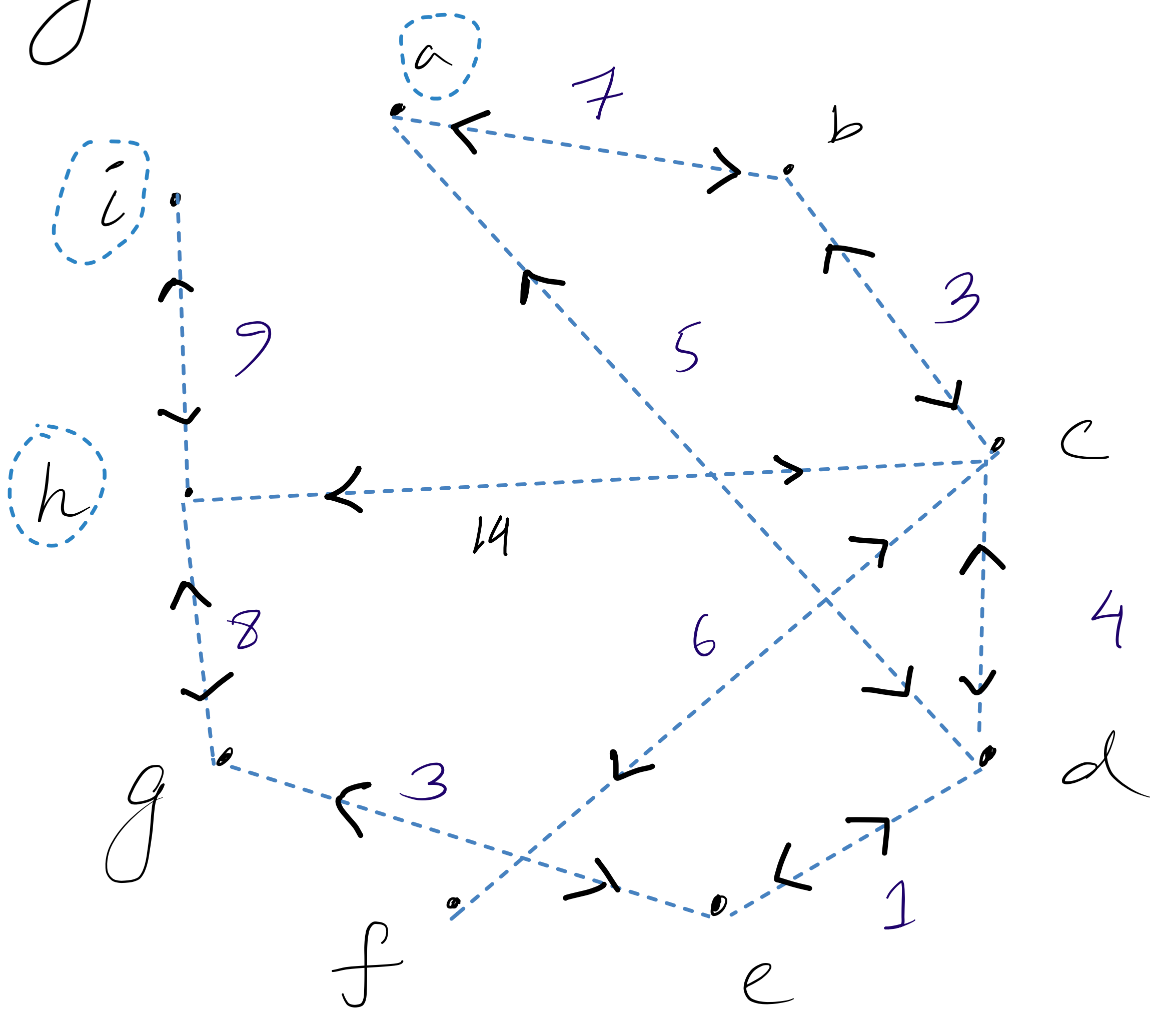
Steps:

1) Some nodes are present which must be present in the path.

'a' and 'i' are obviously included.

'i' node only one connected node, which

is the 'h' node, therefore it is also the part of the shortest path



step => circling the mandatory nodes

2) As, we are dealing with shortest path, any node - node combination which needs to be repeated must be ignored. i.e [contains a terminal node]

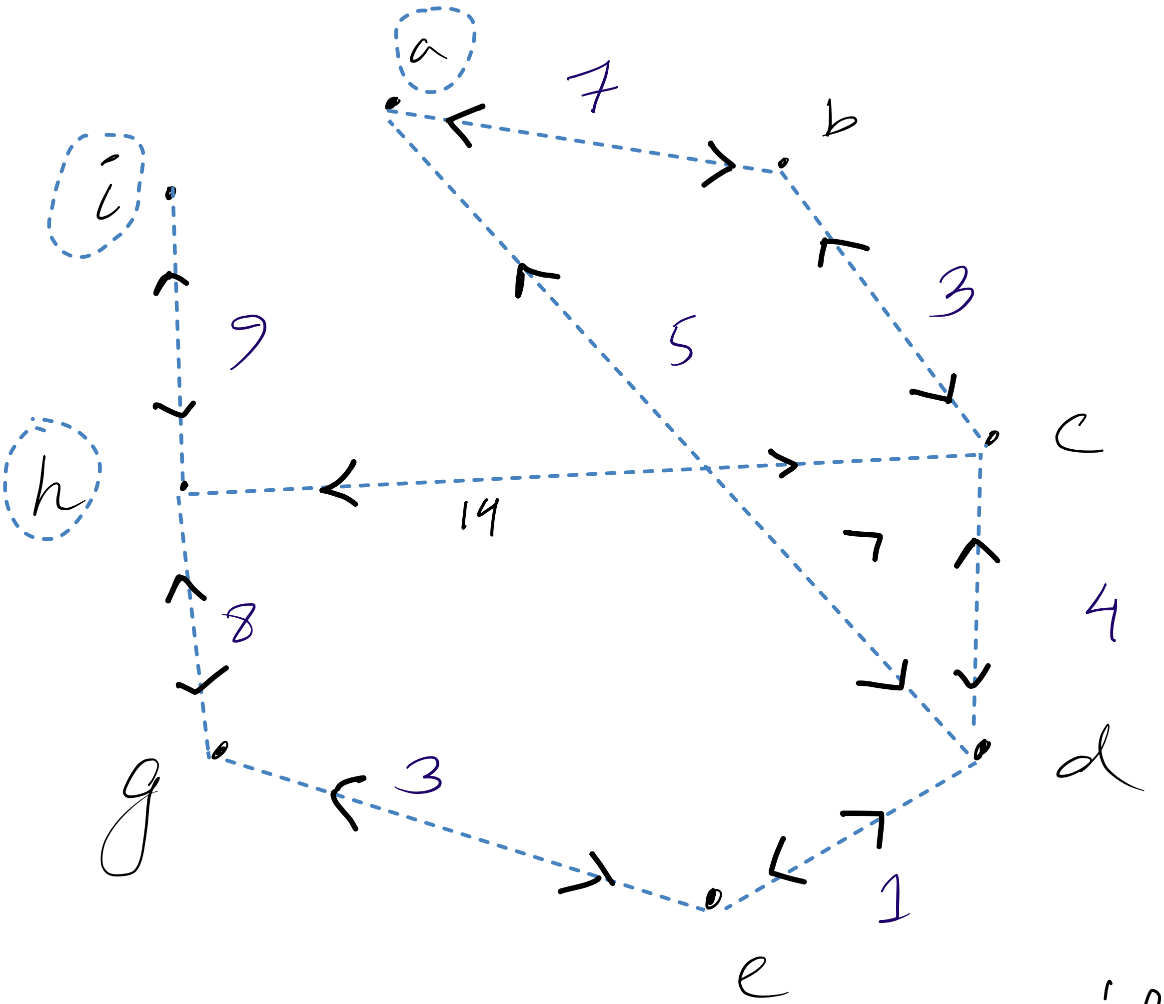
Terminal nodes are :- 'i' 'f'

To travel from 'a' to 'i' using 'f'.
we must go to $c \to f \to e$ which is a

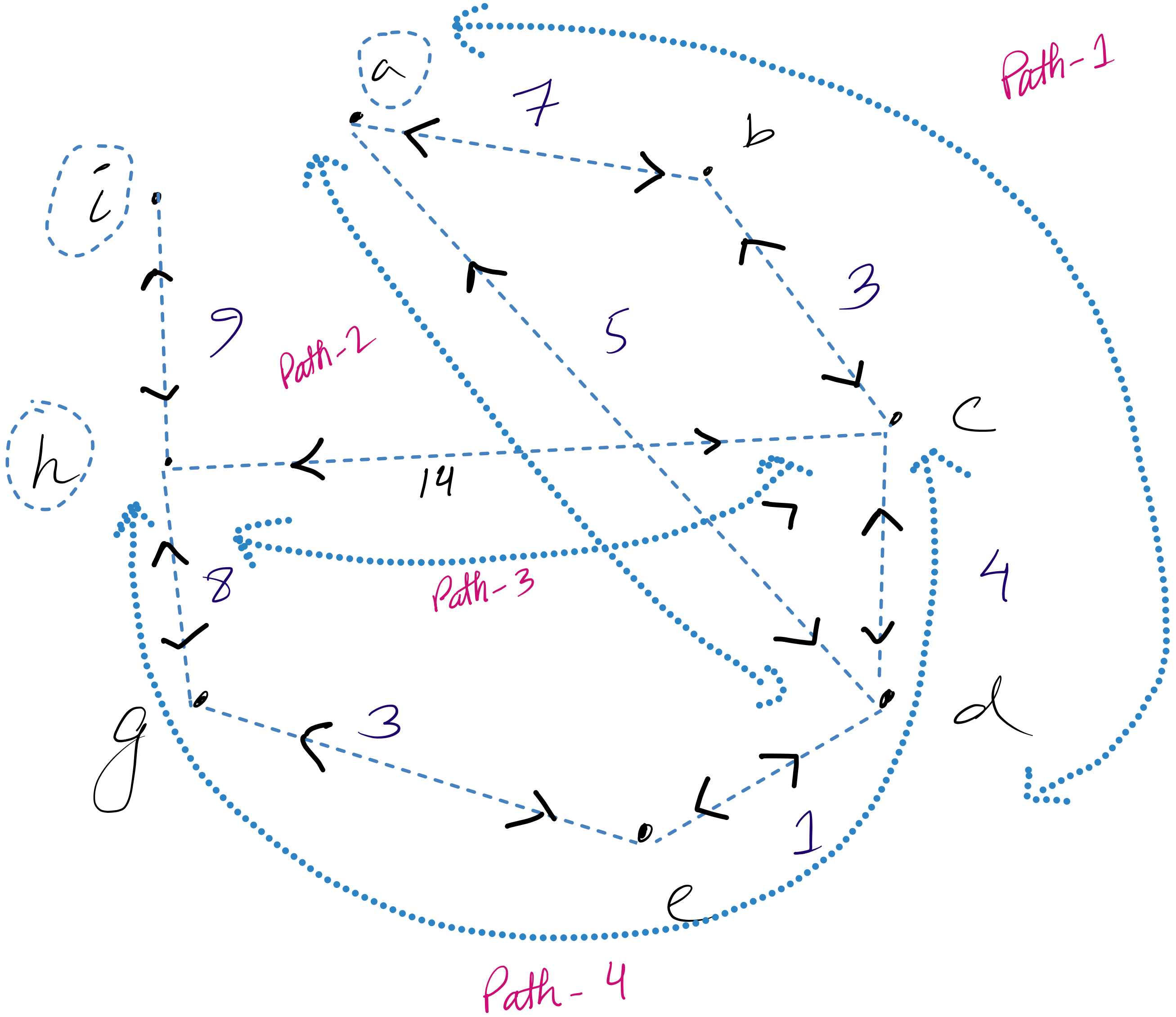
back

reduntant step as we only go back to where we started from.

['f' is eliminated]



Step ⇒ eliminating reduntant nodes 'f'



Step- Comparing 'equivalent' paths

Path-1

1) Travels from a⟺d
2) weight /Length
⟹ 7+3+4 = 14

Path-2

1) Travels from a⟺d
2) weight/Length
⟹ 5

So, Path-1 is redundant and so are its nodes R_1 (b, c) in travelling from a⟺d

Path - 3

1) $c \Leftrightarrow h$

2) Length → 14

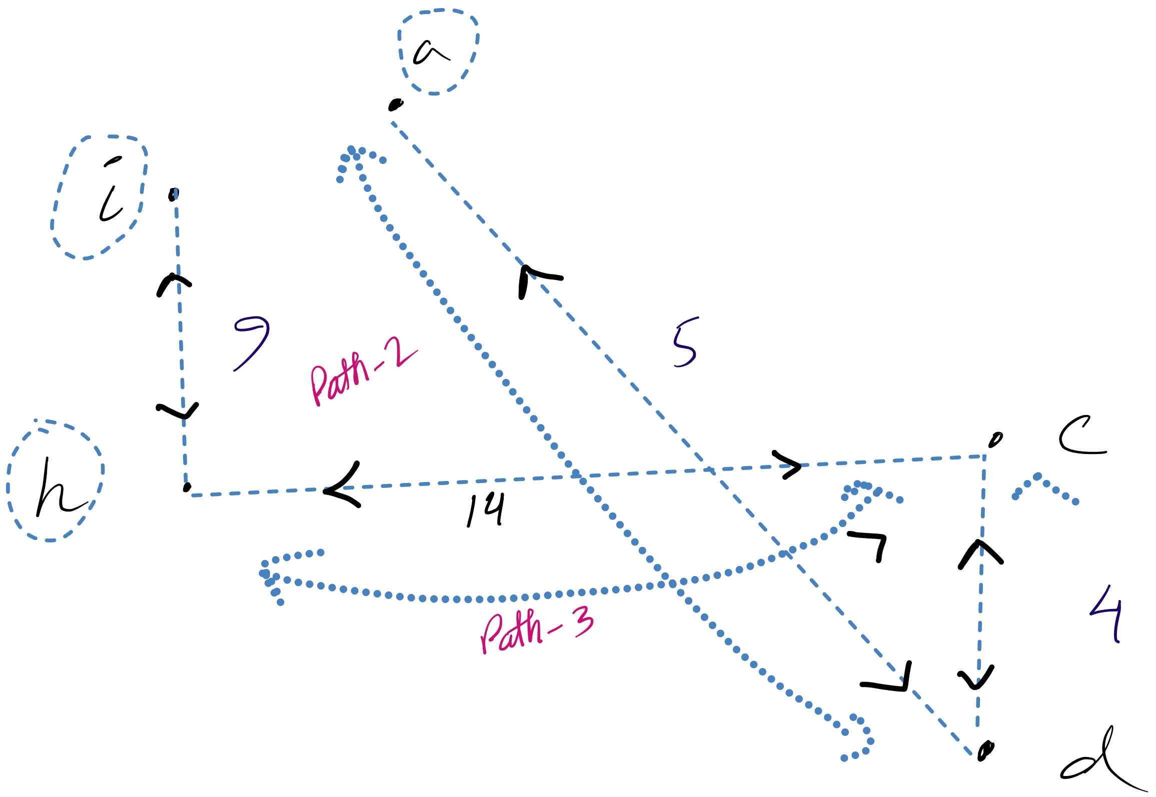
Path - 4

1) $c \longleftrightarrow h$

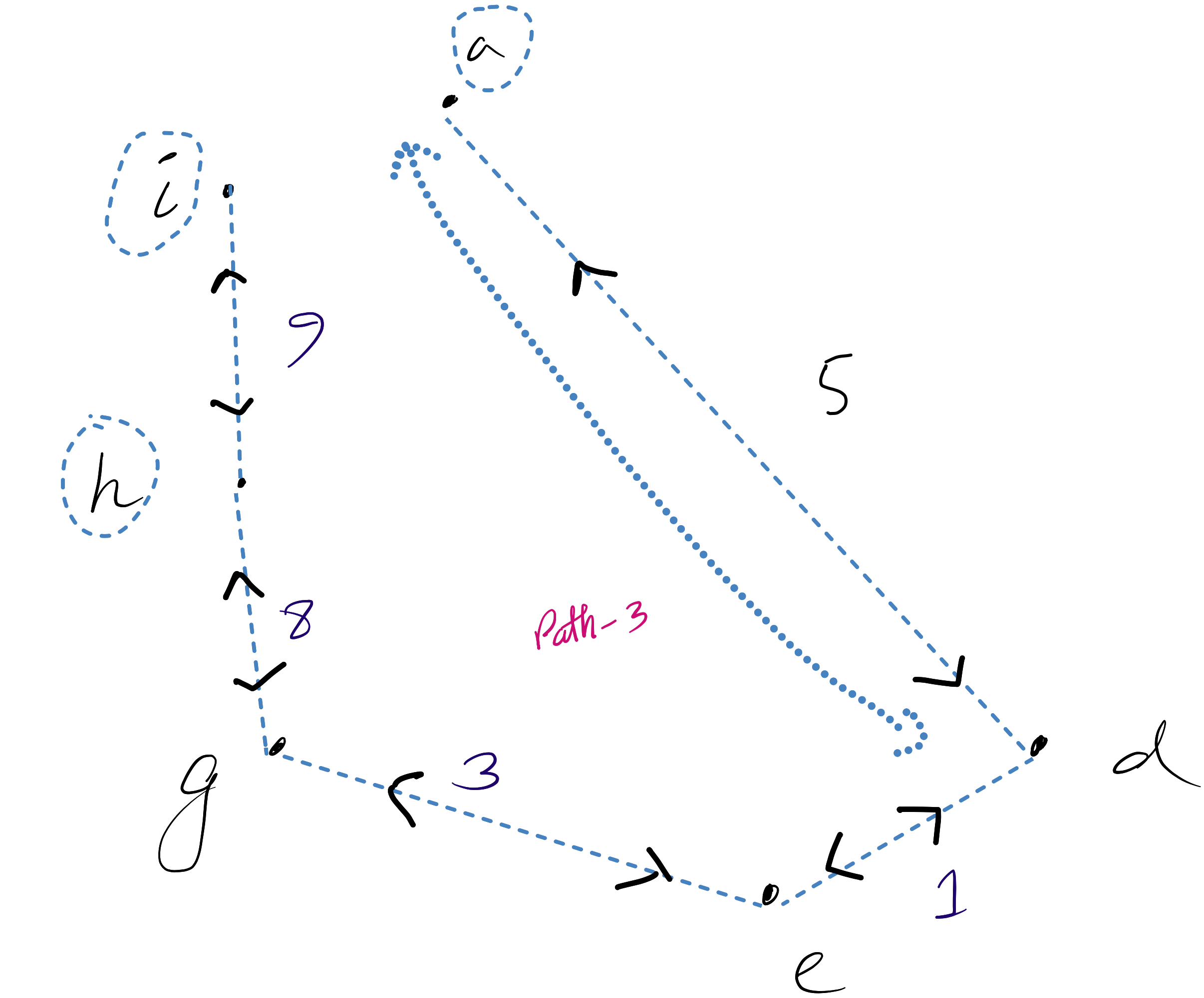
2) Length →
$8+3+1+4 \Rightarrow 16$

Path - 4 is redundant in $c \longleftrightarrow h$

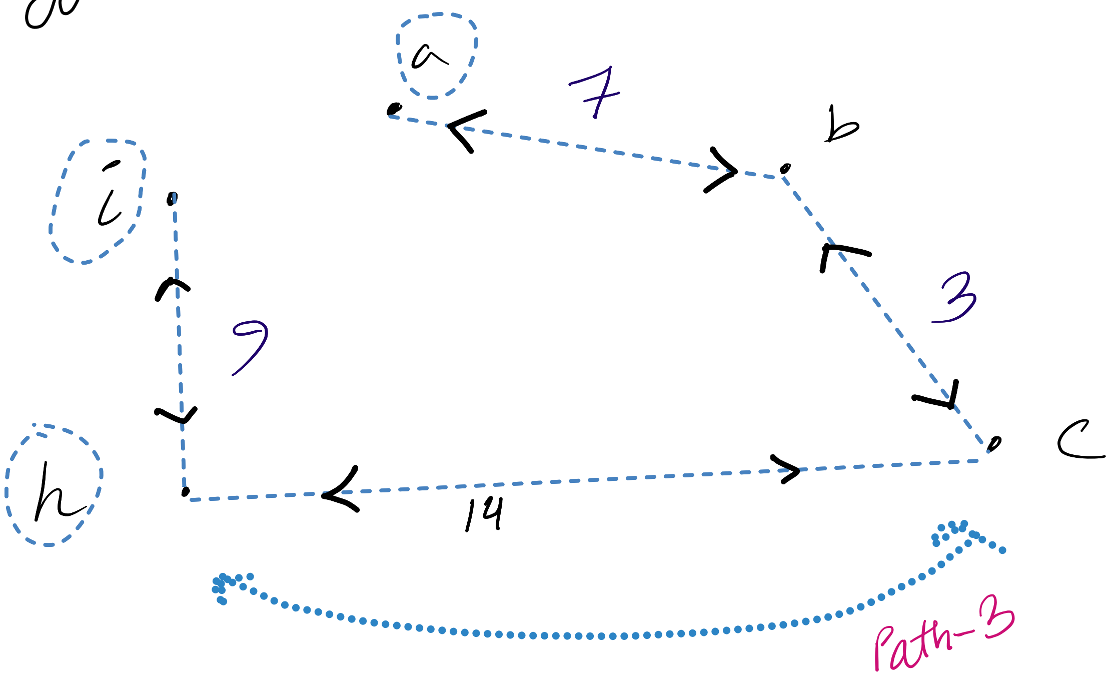
R_2 (e,g) in $c \Leftrightarrow h$



Efficient path - 1



Efficient path-2



Efficient path - 3

Step ⇒ compare unique linear paths

path - 1 = 5+4 + 14+9 = 32

path - 2 = 5+1+3+8+9 = 26

path - 3 = 7+3+14+9 = 33

∴ Most Efficient path ⇒ path - 2

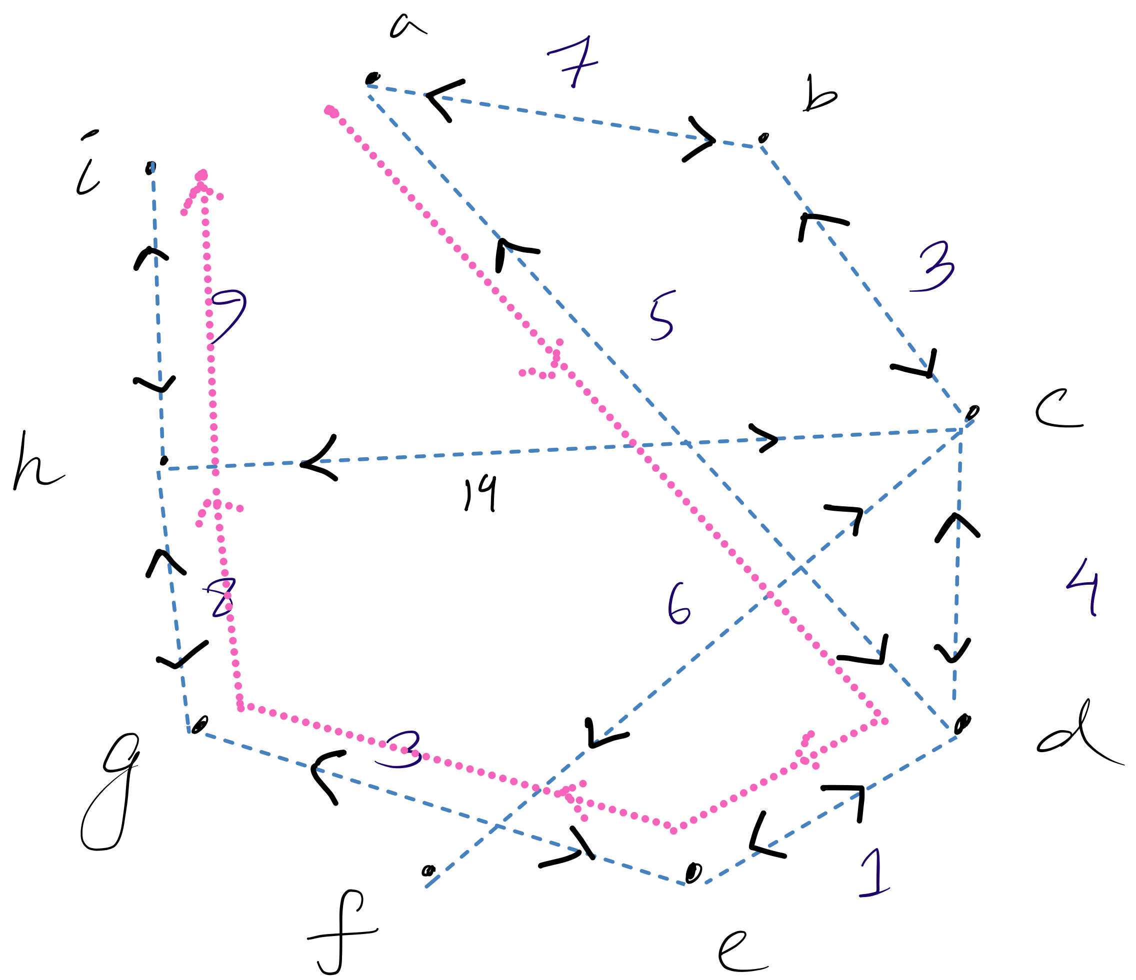


Fig: Dotted line shows the shortest path from node 'a' to node 'i'.

The path ⇒ a d e g h i , The distance = 26