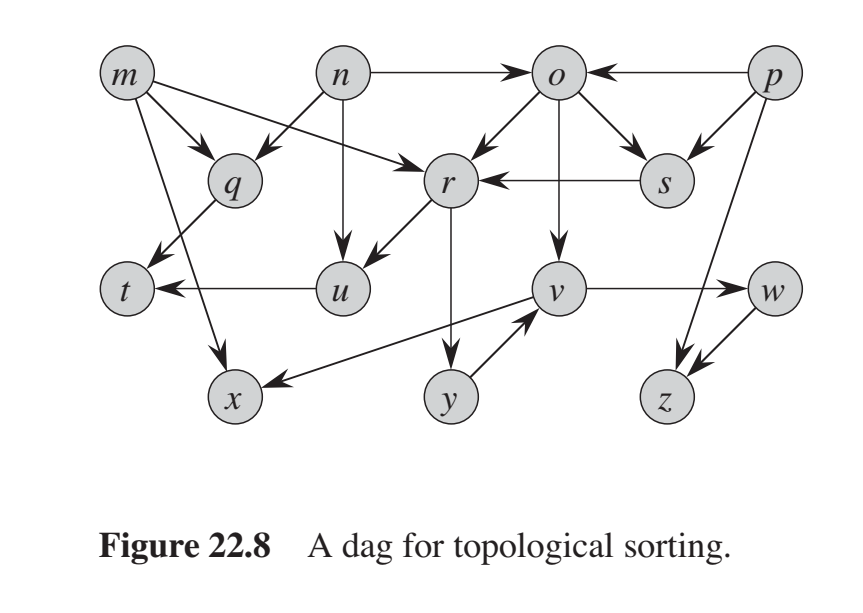
## Topics: Topological sort, strongly connected components, Dijkstra, Bellman-Ford, Prim, Kruskal

**1.**

****

**Perform** topological sort on the above graph. Always choose the alphabetically smaller vertex when there is a choice. Begin DFS from the node m.

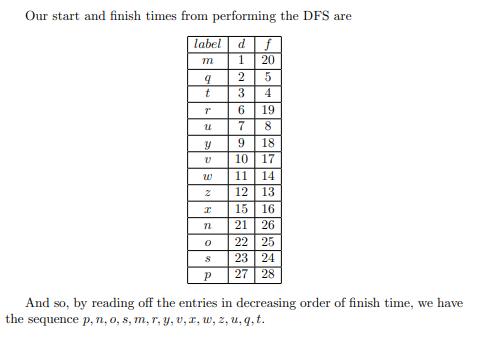
Table for DFS

| Node | Starting time | Finishing time |
| --- | --- | --- |
| m |  |  |
| n |  |  |
| o |  |  |
| p |  |  |
| q |  |  |
| r |  |  |
| s |  |  |
| t |  |  |
| u |  |  |
| v |  |  |
| w |  |  |
| x |  |  |
| y |  |  |
| z |  |  |

Finally, write the topologically sorted order:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Solution



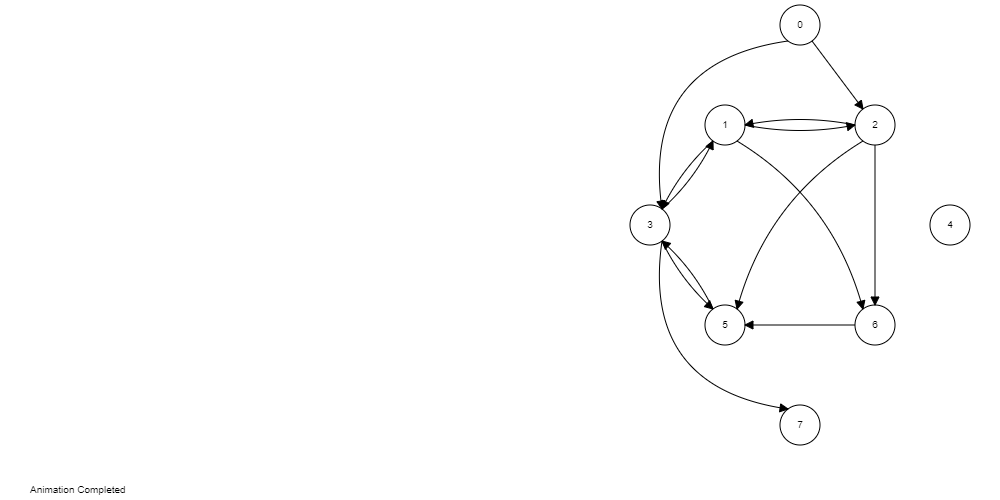
2. **Find** the strongly connected components from this graph. 

Table for DFS (Consider numerically smaller vertex when there is a choice)

| Node | Starting Time | Finishing Time |
| --- | --- | --- |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |

Draw the transpose graph

|  |
| --- |

Table for DFS on transpose graph (Consider vertices in order of decreasing finishing time of first DFS)

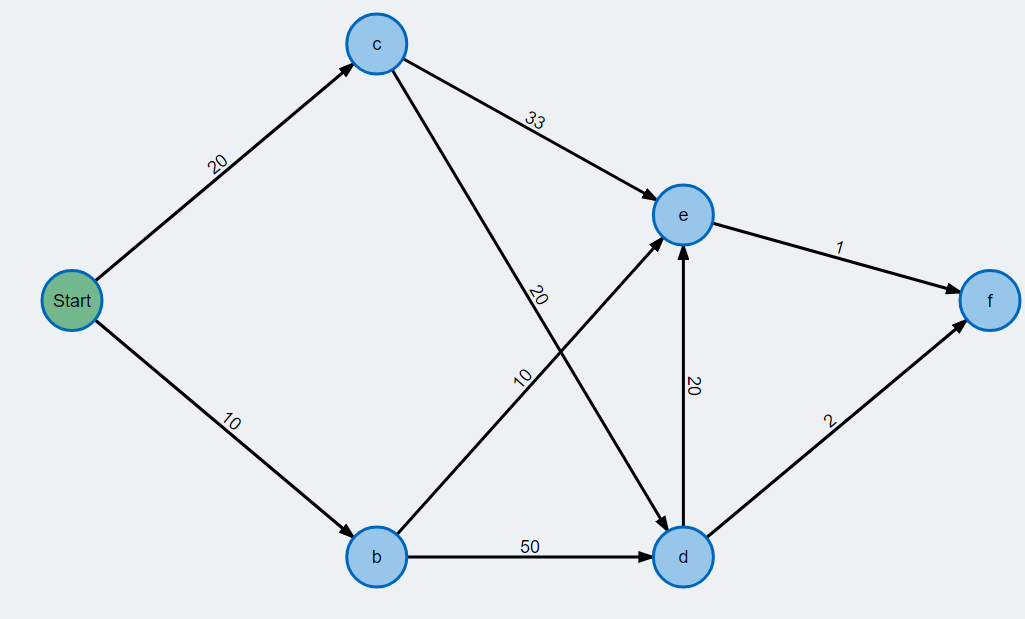
| Node | Starting Time | Finishing Time |
| --- | --- | --- |
| 0 |  |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |

Finally, write the strongly connected components (e.g. : 1-2-3, 4-5, 6 [three strongly connected components] )

Solution

4, 0, 2-1-3-5-6, 7

3.

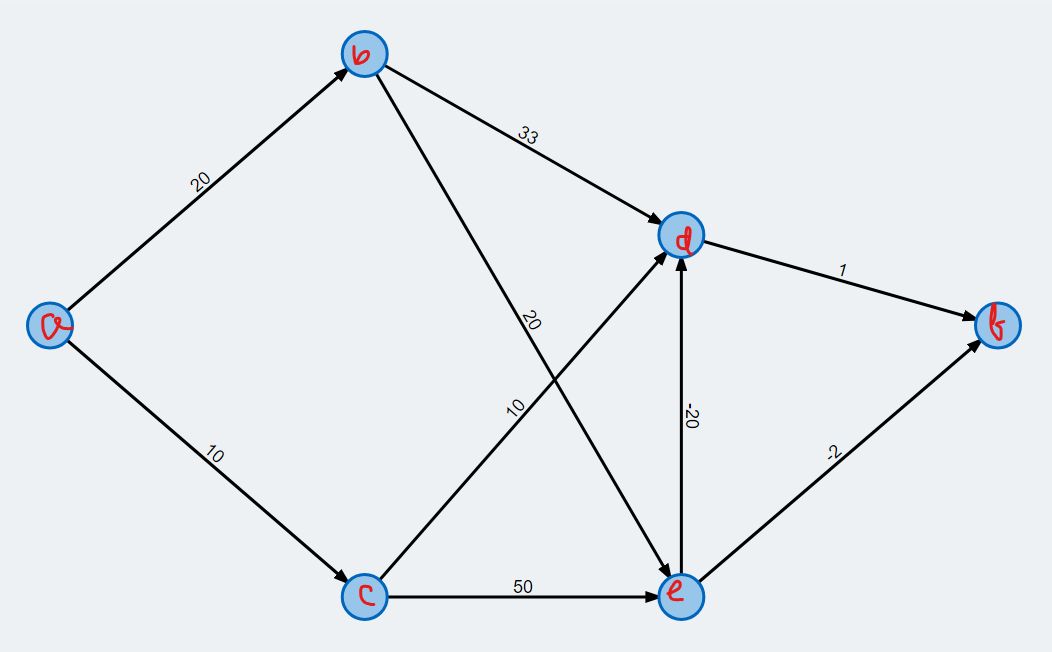


The start node is a.

Perform Dijkstra’s algorithm on the above graph and fill up the following table with the distance values of each vertex. In case of a choice between the same minimum nodes, always choose the alphabetically smallest node.

| Iteration | ExtractMin() | A | B | C | D | E | F |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |

4.

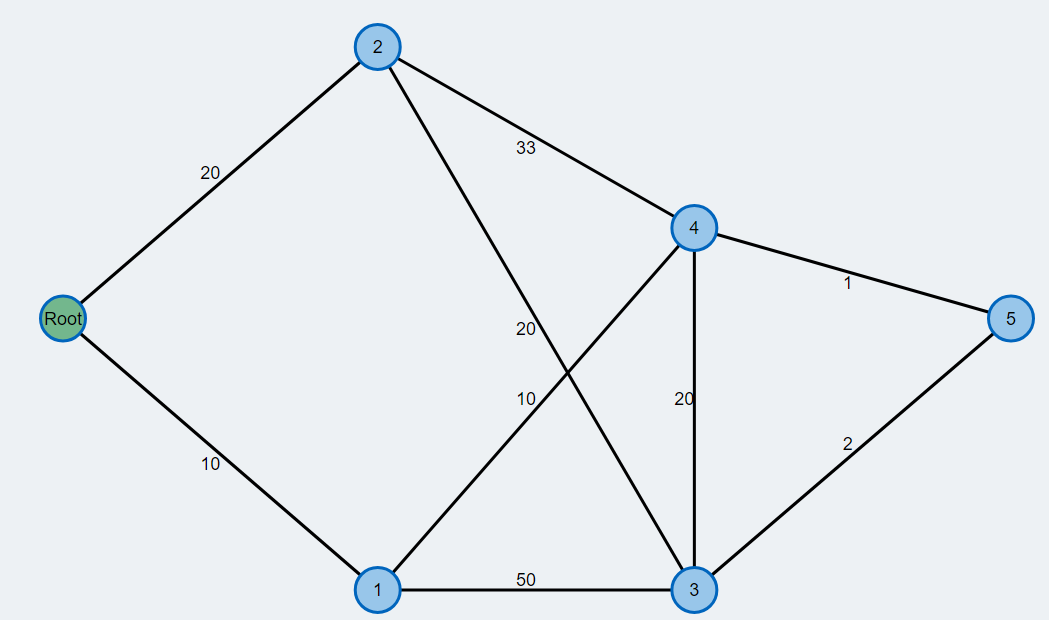


The start node is a.

Perform Bellman-Ford algorithm for one iteration on the above graph and fill up the following table with the distance values of each vertex.

| A | B | C | D | E | F |
| --- | --- | --- | --- | --- | --- |
| 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

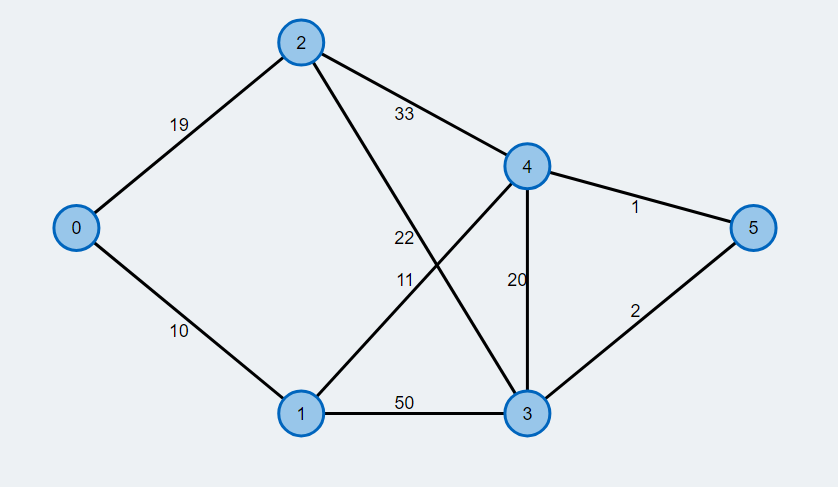
5.



Fill up the following table with key values of each vertex using Prim’s algorithm and draw the final MST. In case of a choice between the same minimum nodes, always choose the numerically smallest node.

| Iteration | ExtractMin() | Root | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |

6.



Complete the following table using Kruskal’s algorithm. Write the edges in non-decreasing order of their weights. Put cross (⨯) beside the edges that are not included in the MST and check (✔) beside the edges that are included in the MST.

| ⨯/✔ | Edges (u,v) | Weights w(u,v) | Sets |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |