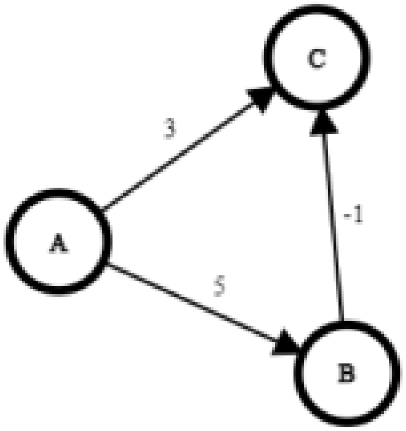
Greedy and DP:

1. How does dynamic programming have a lower time complexity than brute force or recursive solutions? Explain briefly.
2. Explain what you understand by the the value mat[i, j] = x? (Let's assume there will be given a DP table, ask you to explain the meaning of a particular cell.)
3. Explain how greedy property is used in Huffman coding and how it provides an optimal code.
4. Two friends are tasked with building the given Huffman tree. While both friends use priority queues in the process,
    - friend 1 inserts the newly created nodes at the end of the same frequency nodes.
    - friend 2 inserts the newly created node at a random position between the same frequency nodes.

   Explain which technique will provide more optimal code and why. If they are bound to provide equally optimal code, explain why.
5. Can merge sort be written as a dynamic programming algorithm? Explain your answer.
6. In LCS using DP, the space complexity is O(n*m). Can this be further optimized? If yes, then how?
7. In the huffman tree, why do we connect the leaf nodes with lower frequency earlier and greater frequency later. Explain briefly.
8. What is the prefix rule in huffman coding? Explain with example. How does this rule prevent ambiguity among the code of the multiple characters.
9. What is the optimal substructure property in the context of the LCS problem?
10. Can you think of a dynamic programming solution to find the shortest path? (Ans: Bellman ford, since Bellman ford is a DP algorithm)

Graph:

11. You have this graph where there is a negative weight on one edge. Explain why Dijkstra's shortest path algorithm will find correct answers on this graph given node A as the source. Then propose a change in the graph to make the result incorrect.



12. What would happen if we used array instead of priority queue in dijstra algorithm. Explain in respect of time complexity.
13. Why does Bellman ford runs |v| - 1 times? Explain the reading behind this.
14. Why does a topological sort not exist for graphs with cycles?
15. What does it mean if a graph has zero in-degree vertices during topological sort?
16. Can a graph have more than one valid topological ordering? Under what condition?
17. Why do we need two passes of DFS in Kosaraju's algorithm?