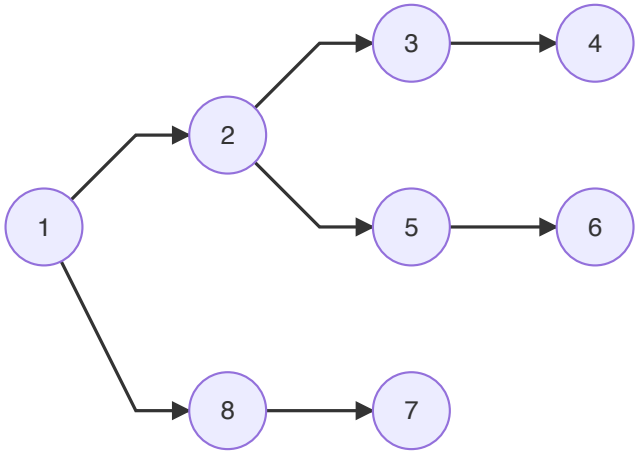
150 Graph and Algorithm 2020 Exam Sample Solution

Disclaimer: This is not an official answer key, so please correct any mistake if you find one. There are more than one possible solution for some questions, so keep in mind about that!

1. **a. i)** Entrance order: 1 2 3 4 5 6 8 7

   Exit order: 4 3 6 5 2 7 8 1

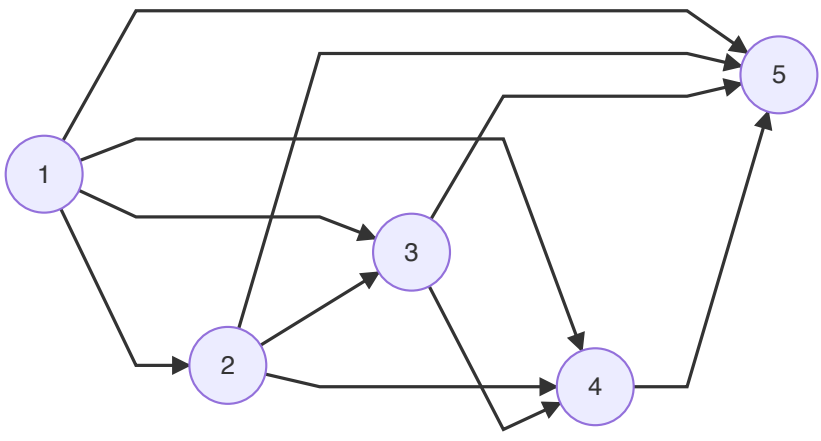
   Topological sort: 1 8 7 2 5 6 3 4 (reverse of the exit order)



ii) Maximum number of arcs: $\frac{n(n-1)}{2}$.

Assume a directed acyclic graph has n nodes. We take the first node, which can have up to $n-1$ directed arcs pointing to other nodes. Then take the second node, which can have up to $n-2$ directed arcs pointing to nodes other than itself and the first one(since the graph is acyclic, it cannot connect to the first node). This pattern goes on until the last node. Hence, in total, we can have up to $(n-1)+(n-2)+(n-3)+\ldots+1+0=\frac{n(n-1)}{2}$ arcs in this directed acyclic graph. (Thanks to Fankai Yan for noticing the typo here).

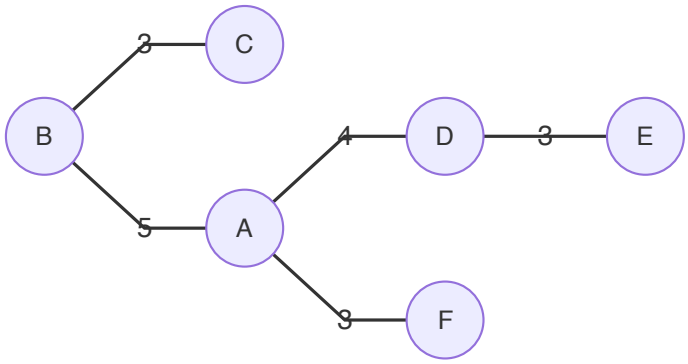
5-nodes DAG with maximum number of arcs:



b. Since the graph is simple, we assume that it does not contain loops or parallel arcs. Suppose the graph is connected and has n nodes: each node could have degree between 1 to $n-1$. There are $n-1$ distinct numbers in the range from 1 to $n-1$, but there are n edges in total. We cannot map each number to each node and there would always be one node left out. Hence, there must be at least one pair of nodes that have the same degree.

If the graph is disconnected, then we can use this argument inside each connected subgraphs to prove the assertion.

c. i) The corresponding spanning tree:

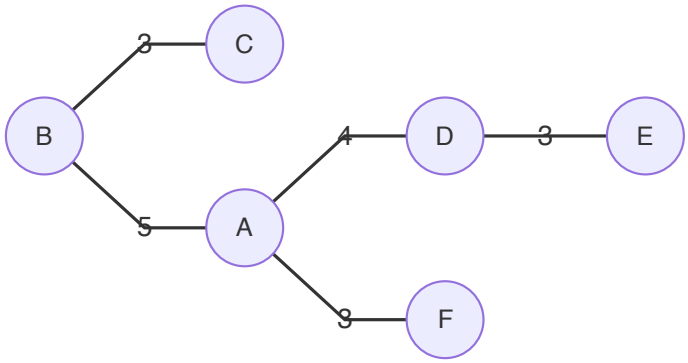


Shortest path length: 7

Order of node traversal: A F D B E C (*Notice C could be optional since once we found E, we can stop searching and return*)

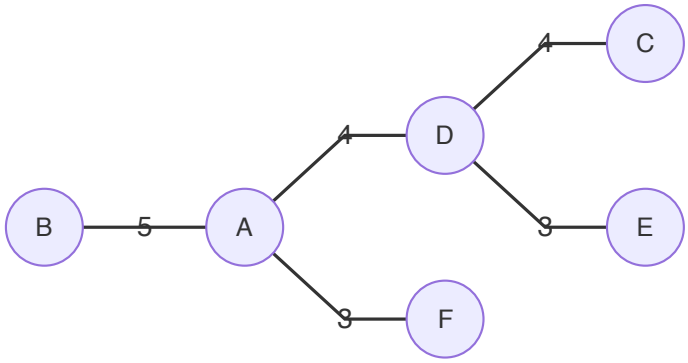
ii) The heuristic being consistent means that for any adjacent node x, y, $h(s) \leq W(x,y) + h(y)$ and $h(\text{finish} = 0)$.

iii) The corresponding spanning tree:



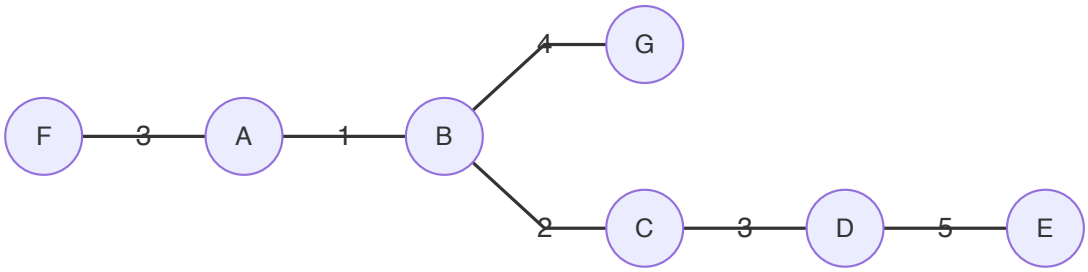
Order of traversal: A F D B E C

Alternatively:



Order of traversal: A F D E B C

2. a. i) The MST:



Order of nodes: A B C F D G E

ii) It does have a unique MST since there is no other selections of arcs that would have the same weight as the MST in i). e.g. when trying to include node E, there's no other arc that also has weight 5.

b. i) 2 swaps at least. (assuming that "sort" means from smallest to largest)

ii) $n-1$ times at least

iii) Because after the first element, each element is one position to the right from where they are supposed to be. This requires moving them one by one with total of $n-1$ moves.

c. i) It means that there is a p-time computable function such that $D(x)$ will hold if and only if $D'(f(x))$ will hold.

ii) $HPA \in NP$. This is true because we can find a corresponding verification problem $\mathrm{VeriHPA(G,}\pi\mathrm{)}$ where G is the input graph and π is a full list of nodes in some order from G containing x and y that are joined by arc a. $\mathrm{VeriHPA(G,}\pi\mathrm{)}$ will be true if traversing the graph G using the order of nodes in π would form a Hamiltonian path. Clearly, $\mathrm{HPA(G,\ a)} \Leftrightarrow \mathrm{VeriHPA(G,}\pi\mathrm{)}$. The size of the list π is clearly bounded by the number of nodes in n, hence HPA is in NP.

iii) We have shown that $\mathrm{HPA} \in \mathrm{NP}$. To show that $\mathrm{HPA} \in \mathrm{NPC}$, we need to show that $\mathrm{HamPath} \leq \mathrm{HPA}$ as well. To show such reduction is possible, we need to find a reduction function f that is p-time computable.

The reduction function f could be as follows: take the original graph G and add two new nodes x and y. Let x connects to all the nodes in G and y connects to x. Let the arc connecting y to x be a. Then we have $\mathrm{HamPath(G)} \Leftrightarrow \mathrm{HPA}(f(G),\mathrm{a})$ since if G has a Hamiltonian path, $f(G)$ must also have a Hamiltonian path using a in order to include y.

Clearly, f is p-time computable since adding two new nodes and connecting them are constant operations, and connecting x to all nodes in G requires n steps. Those operations are clearly polynomial.