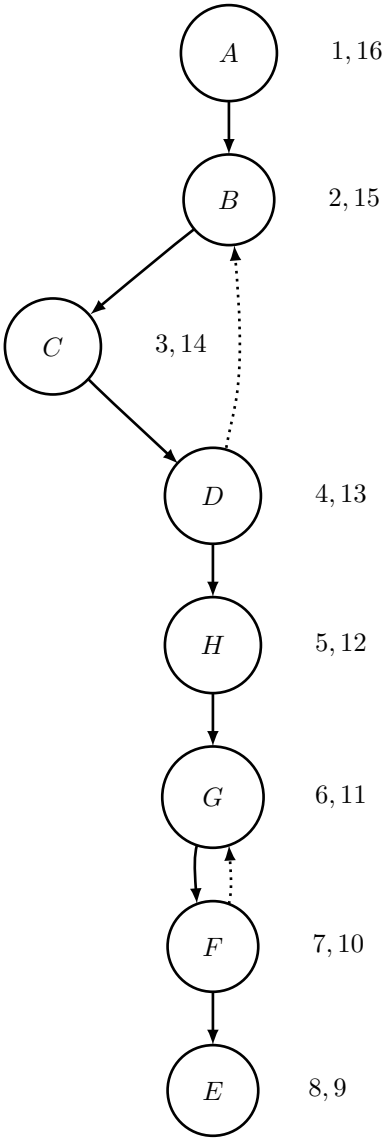
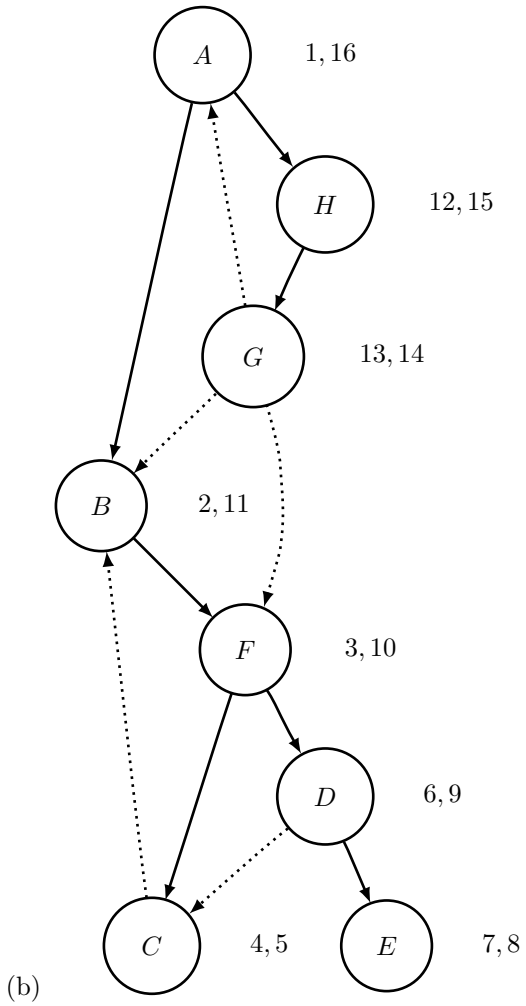
CS 3510 - Spring 2009
Homework 2
Due: March 2

1. Problem 3.2 from [DPV]

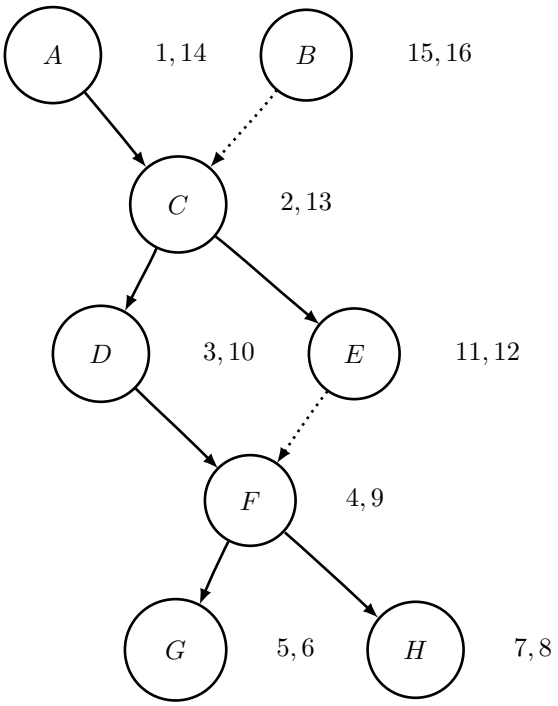


(a)



(b)

2. Problem 3.3 from [DPV]



(a)

(b) The sources are A and B and the sinks are G and H.

(c) B, A, C, E, D, F, H, G

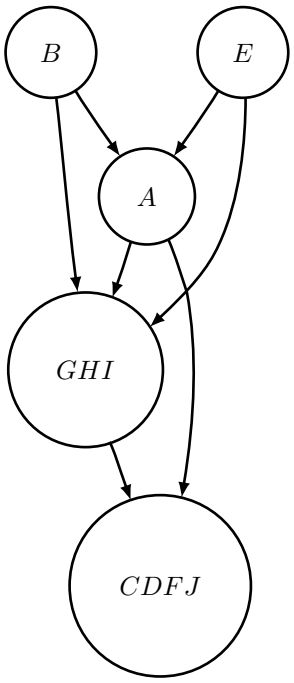
(d) There are two choices for position 1, A or B, which fixes position 2. Position 4 can be D or E, which again decides what will be in position 5. Finally, position 7 has a choice of two nodes, G or H, which fixes the final position. So there are 2*2*2=8 valid combinations for topological orderings.

3. Problem 3.4 from [DPV]

(a) i. {G, I, H}, {F, J, C, D}, {A}, {E}, {B}

ii. Sink: {F, J, C, D}
Source: {E}, {B}

iii.

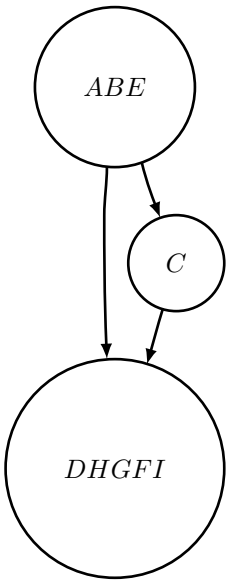


iv. Two. If we add an edge from {F, J, C, D} to {B} and {E}, the graph will be strongly connected.

(b) i. {D, H, G, I, F}, {C}, {A, B, E}

ii. Sink: {DHGFI}
Source: {ABE}

iii.



iv. One. If we add an edge from {D, H, G, F, I} to {A, B, E}, the graph will be strongly connected.

4. Problem 3.12 from [DPV]

   This statement is true. The edge $e = \{u, v\}$ may be one of three types of edges: tree, back, or cross. If e is a tree edge and $post(u) < post(v)$ then v is the ancestor of u. Once the subgraph containing u has been exhausted, DFS will return to the ancestor of u and so it will have a higher post number than u. Therefore, v is the ancestor of u in this case.

   If e is a back or cross edge, then e will be an edge in G, but not in the DFS tree. This is because this edge refers to a node that has already been explored. Since $post(u) < post(v)$, this explored node is u. Since u has already been explored before the termination of the DFS of v, then it must be a descendant of v.

   Therefore v is the ancestor of u in the DFS when $post(u) < post(v)$.

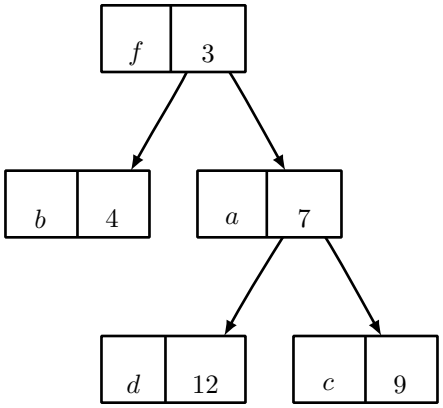
5. *Global Sink*
   First we run DFS on G^R.The explore subroutine can only be called one time. Otherwise, it means that the source strongly connected components cannot reach some subset of nodes in the graph. This corresponds to nodes in the original graph being unable to reach the global sink, which disqualifies it as a global sink.

   If this is not the case, the strongly connected component with the highest post number is a source strongly connected component that is able to

reach all other nodes in the graph. So all nodes in the original graph are able to reach the global sink, satisfying the second property. If there were any other component with this property then all of its nodes would have a path to and from the previous component, which would make it a part of the same strongly connected component.

Additionally, this source strongly connected component cannot consist more of a single node because this would mean that the 'global sink' has an outgoing edge violating the first property. So this node is the global sink. Since this algorithm only uses DFS, it has the same running time $O(|V| + |E|)$.

6. *Binary heap*



7. Problem 4.19 from [DPV]

Since the weights contributed by the nodes are positive, we can use a modified Dijkstra's to solve this problem. The cost of crossing an edge is no longer just l_e, it now includes contributions from the endpoints u and v. If we assigned each edge a cost of $l_e + c_u + c_v$, the weight of all interior nodes of a path would be counted twice. Instead we replace all instances of $dist(u) + l(u, v)$ with $dist(u) + l(u, v) + c_v$. This way, $dist(u)$ will equal the total cost of a path so far, including the current node and its contribution will only be added once. Also note that the line $dist(s) = 0$ will be replaced by $dist(s) = c_s$. Now the dist array returned by Dijkstra's will be exactly the cost array.

8. Problem 4.21 from [DPV]

   (a) We want to define a graph with nodes for every currency and an edge cost that corresponds to the loss inccured by a currency conversion. The percent value lost or $1-r_{i,j}$ can serve this purpose. This heuristic will allow us to choose the edge that will yield the lowest reduction in value at a given node. Then we can run Dijkstra on this graph to find the best sequence of conversions for any two currencies.

   (b) Using the graph representation of part a, in order for there to be profit, one of the rates in a conversion path would be greater than 1. This means that the corresponding edge in the graph would be negative. So to detect this event, we need to find a negative edge and this can be done in linear time with depth first search.