1. Graph Representations

(a) Write the graph above as an adjacency matrix, then as an adjacency list. What would be different if the graph were undirected instead?

(b) Give the DFS preorder, DFS postorder, and BFS order of the graph traversals starting from vertex A. Break ties alphabetically.
2 Dijkstra’s Algorithm

For the graph below, let $g(u, v)$ be the weight of the edge between any nodes $u$ and $v$. Let $h(u, v)$ be the value returned by the heuristic for any nodes $u$ and $v$.

(a) Run Dijkstra’s algorithm to find the shortest paths from $A$ to every other vertex. You may find it helpful to keep track of the priority queue and make a table of current distances.

**Pseudocode**

1. $PQ = \text{new PriorityQueue}()$
2. $PQ.add(A, 0)$
3. $PQ.add(v, \infty)$ # (all nodes except A).
4. 
5. $\text{distTo} = \emptyset$ # map
6. $\text{distTo}[A] = 0$
7. $\text{distTo}[v] = \infty$ # (all nodes except A).
8. 
9. while (not $PQ.$isEmpty()):
   10.       poppedNode, poppedPriority = $PQ.$pop()
   11.       
   12. for child in poppedNode.children:
           if $PQ.$contains(child):
                       potentialDist = $\text{distTo}[\text{poppedNode}] + \text{edgeWeight}(\text{poppedNode}, \text{child})$
           if potentialDist < $\text{distTo}[\text{child}]$:
                       $\text{distTo}.$put($\text{child}, \text{potentialDist}$)
                       $PQ.$changePriority($\text{child}, \text{potentialDist}$)
(b) Given the weights and heuristic values for the graph below, what path would
A* search return, starting from A and with G as a goal?

**Pseudocode**

```plaintext
1 PQ = new PriorityQueue()
2 PQ.add(A, h(A))
3 PQ.add(v, infinity) # (all nodes except A).

4 distTo = {} # map
5 distTo[A] = 0
6 distTo[v] = infinity # (all nodes except A).

8 while (not PQ.isEmpty()):
  9 poppedNode, poppedPriority = PQ.pop()
10 if (poppedNode == goal): terminate

13 for child in poppedNode.children:
  14 if PQ.contains(child):
  15 potentialDist = distTo[poppedNode] + edgeWeight(poppedNode, child)
  16
  17 if potentialDist < distTo[child]:
  18 distTo.put(child, potentialDist)
  19 PQ.changePriority(child, potentialDist + h(child))
```

(c) Is the heuristic admissible? Why or why not?