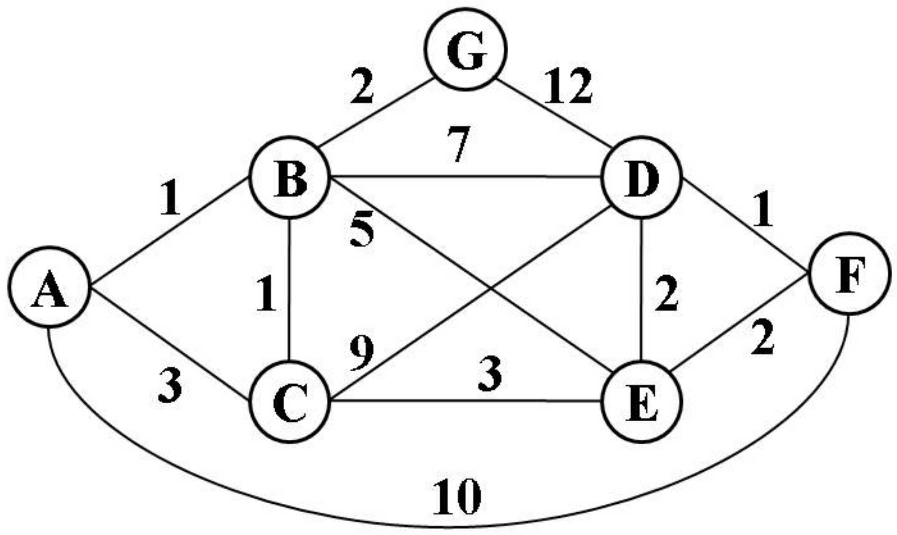
CSE373 Fall 2013

Example Exam Questions on Dijkstra's Algorithm (and one on Amortized Analysis)

Name:____

1. Consider the following undirected, weighted graph:



Step through Dijkstra's algorithm to calculate the single-source shortest paths from A to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them known. Finally, indicate the lowest-cast path from node A to node F.

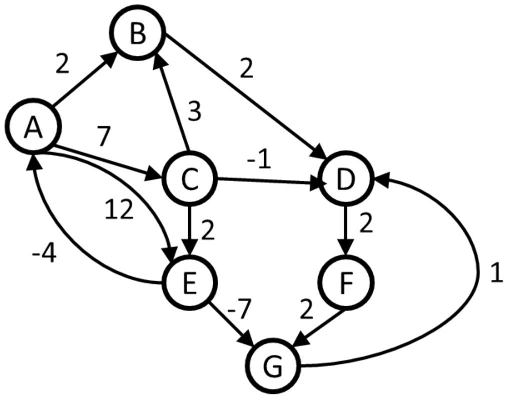
Solution:
Known vertices (in order marked known): A B C G E D or F D or F

Vertex	Known	Cost	Path
A	Y	0	
B	Y	1	A
C	Y	3 2	A B
D	Y	8 7	B E
E	Y	6 5	B C
F	Y	10 7	A E
G	Y	3	B

Lowest-cost path from A to F: A to B to C to E to F

Name:____

2. Consider the following directed, weighted graph:



(a) Even though the graph has negative weight edges, step through Dijkstra's algorithm to calculate *supposedly* shortest paths from A to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them known.

Solution:
Known vertices (in order marked known): A B D F C G E

Vertex	Known	Distance	Path
A	Y	0	
B	Y	2	A
C	Y	7	A
D	Y	4	B
E	Y	12 9	A C
F	Y	6	D
G	Y	8	F

(b) Dijkstra's algorithm found the wrong path to some of the vertices. For just the vertices where the wrong path was computed, indicate *both* the path that was computed and the correct path.

(c) What *single* edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?

Solution:

(b) Computed path to G is A,B,D,F,G but shortest path is A,C,E,G.
Computed path to D is A,B,D but shortest path is A,C,E,G,D.
Computed path to F is A,B,D,F but shortest path is A,C,E,G,D,F.

(c) The edge from E to G.

Name:____

3. Suppose we define a different kind of graph where we have weights on the vertices and not the edges. Does the *shortest-paths problem* make sense for this kind of graph? If so, give a precise and formal description of the *problem*. If not, explain why not. Note we are not asking for an algorithm, just what the problem is or that it makes no sense.

   Solution:
   Yes, this problem makes sense: Given a starting vertex v find the lowest-cost path from v to every other vertex. The cost of a path is the sum of the weights of the vertices on the path.

Name:____

4. Consider using a simple linked list as a dictionary. Assume the client will never provide duplicate elements, so we can just insert elements at the beginning of the list. Now assume the peculiar situation that the client may perform any number of insert operations but will only ever perform at most one lookup operation.

   (a) What is the worst-case running-time of the operations performed on this data structure under the assumptions above? Briefly justify your answer.

   (b) What is the worst-case amortized running-time of the operations performed on this data structure under the assumptions above? Briefly justify your answer.

   Solution:

   (a) inserts are $O(1)$ (push on the front of a linked list), but the lookup is $O(n)$ where n is the number of inserted items since the lookup may be last and be for one of the earliest inserted items

   (b) amortized all operations are now $O(1)$. Inserts are still $O(1)$. And the lookup can take at most time $O(n)$ where n is the number of previously inserted items. So the total cost of any n operations is at most $O(n)$, which is amortized $O(1)$.