This is a closed book test. You may not refer to any resources.

This is a 50 minute test.

Please write your answers in ink. Pencil answers will be marked, but will not be reconsidered after they have been marked.

The test will be marked out of 50.

<table>
<thead>
<tr>
<th>Question 1</th>
<th>/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 2</td>
<td>/15</td>
</tr>
<tr>
<td>Question 3</td>
<td>/15</td>
</tr>
</tbody>
</table>

**TOTAL** /50
Question 1 (20 marks)

In this question you are required to modify Dijkstra’s Algorithm to solve a new problem.

Here is Dijkstra’s Algorithm, much as presented in class. The input is a graph $G$ consisting of vertices and edges with numeric weights on the edges, and a start vertex $A$. We use a queue $Q$ to keep track of vertices awaiting processing, and indexed structures (such as arrays) $P$, $D$, and $M$ to keep track of Predecessors, Distances, and Marked vertices (vertices are marked when we remove them from $Q$).

1. $DA(G,A)$:
   2. $D[A] = 0$
   3. for all vertices $x$ other than $A$:
   4. $D[x] = \text{infinity}$
   5. $M[x] = \text{“no”}$
   6. add $A$ to $Q$
   7. while $Q$ non-empty:
   8. let $x$ be the vertex in $Q$ that has the smallest $D$ value
   9. remove $x$ from $Q$
   10. $M[x] = \text{“yes”}$
   11. for each $y$ that is a neighbour of $x$ with $M[y] == \text{“no”}$:
   12. if $D[x] + \text{weight}(x,y) < D[y]$:
   13. $D[y] = D[x] + \text{weight}(x,y)$
   14. $P[y] = x$
   15. if $y$ is not in $Q$:
   16. add $y$ to $Q$
17. return $P$  #$P$ contains the path information

Now consider this problem: Suppose we have a graph in which the vertices represent banks and the edges represent money-transfer agreements. - so an edge between vertex $X$ and vertex $Y$ means that Bank $X$ is willing to transfer money to Bank $Y$. The banks are evil and skim off a percentage of each transfer – the percentage that is actually transferred on each edge is given as the edge-weight. If money is transferred along a multi-step path, the actual percent of the original amount that goes through to the end of the path is the product of the percentages on the edges in the path.

A customer who has her money in bank $A$ wants to know the best paths along which to transfer her money to other banks – that is, the highest-value paths from $A$ to all other vertices.

The next page shows a small example to demonstrate the problem.
In this graph there are two paths from A to B. Going through C, the first step reduces the amount to 80% and the second step reduces that amount to 50% of the already reduced value – so the value at the end is just 40% of the original value (80% * 50%). Going through D, the first step reduces the amount to 70% and the second step drops it to 90% of that, giving a final value of 63% of the original. In this example, going through D is better because 63% > 40%.

Modify Dijkstra’s Algorithm to solve this problem. Please indicate your changes using the line numbers given in the statement of the algorithm shown above, such as “Change Line 7 to ...”

You only need to change 5 lines.
QUESTION 2 (15 marks)

Suppose X and Y are both problems in the class NP. X is known to be NP-Complete, but the complexity of Y is unknown.

Each part of this question provides you with more information, and asks you to explain the significance (if any) of the new information. Each part of this question is independent of all the other parts.

(a) Suppose it is proved that X reduces to Y. What (if anything) does this tell us?

(b) Suppose it is proved that Y reduces to X. What (if anything) does this tell us?

(c) Suppose an O(n^{16}) algorithm is found for X. What (if anything) does this tell us?
Question 3 (15 marks)

Given a set of n integers, already sorted, give a Divide & Conquer algorithm that will find the two values that are closest together in value.

For example, in the set {1, 3, 4, 7, 12, 14, 18, 21} the closest values are 3 and 4.

Divide & Conquer is not necessarily the best way to solve this problem. The purpose of this question is to give you the opportunity to demonstrate familiarity with the Divide & Conquer method.