COMP 282 - Final (Fall, 2018) Name:

Question 1 Give the best data structure to use in the following scenarios. Justify your answer. You may assume sufficient system resources to support whichever data structure you choose.

(a) You want to store a large number of integers. They do not change often, but you need to be able to search them quickly.

(b) You need to be able to store an unknown number of floating point values. You only need to access and delete latest value that had been stored.

(c) You want to store a large number of integers. They change often, and you need to search them quickly.

(d) You have a series of bank records, consisting of a name, account number, and balance. You need to be able to quickly insert new records, update a given record, and delete records.

(e) You have a very small number of strings. You do not need to insert or delete strings, but you do need to search them quickly.

Question 2 Use *Djikstra's algorithm* to find all-pairs shortest paths from v_0 to vertices in its connected component in the following graph.

$$V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$$

Edge	Weight
$\{v_0, v_1\}$	4
$\{v_0, v_3\}$	6
$\{v_0, v_6\}$	3
$\{v_1, v_4\}$	2
$\{v_1, v_2\}$	2
$\{v_2, v_3\}$	1
$\{v_2, v_7\}$	4
$\{v_3, v_4\}$	2
$\{v_3, v_5\}$	1
$\{v_3, v_7\}$	7
$\{v_4, v_5\}$	6
$\{v_4, v_6\}$	1
$\{v_6, v_7\}$	4
$\{v_7, v_5\}$	2

Question 3 Given the following tables, and SQL statement, construct a table that represents the results you would expect when running this query.

Customers

CustomerId	Name
1	Bob Roberts
2	Claire McClaireskill
3	Dean VonDeanerson
4	Elanor Elanorsky

Products

ProductId	Name	Price
1	Party Hat	2.25
2	Cat-Sized Party Hat	1.00
3	Fire-Retardant Sparklers	4.00
4	Tactical Pumpkin	4.50
5	Turkey-Flavored Cranberry Sauce	5.00
6	Chimney Security Alarm	15.00

Orders

CustomerId	ProductId	Quantity	Date
2	1	1	2018-01-01
2	2	3	2018-01-01
1	4	15	2018-10-31
4	5	37	2018-11-22
3	6	1	2018-12-25

```
SELECT
  c.Name as Name,
  p.Name as Product,
  o.Quantity as Quantity,
  o.Quantity * p.Price as Total
FROM Customers as c, Products as p, Orders as o
WHERE c.CustomerId = o.CustomerId
  AND p.ProductId = o.ProductId
  AND o.Quantity > 3;
```

Question 4 Construct an *AVL Tree* by inserting the following values in order. Show all intermediate trees, as well as the balance factor calculated for each node.

`1, 2, 3, 7, 8, 5`

Question 5 Build a *Hash Table* of size 8 by inserting the following values in order. Use the algorithm provided to calculate hash values, and open addressing to resolve all collisions. Show all intermediate tables, include both the value and its index in your illustrations.

```
12, 456, 137, 10907, 1144, 953, 4, 3713
```

```
int hash (int input) {
  int h = input % 10;
  h += (input / 10) % 10;
  return h % 8;
}
```

Question 6 Briefly explain the following data structures:

(a) Binary Tree

(b) 2-3-4 Tree

(c) Linked List

(d) Hash Table

(e) Sorted Array

Question 7 Give a definition for the following terms:

(a) Collision Chaining

(b) Adjacent Nodes

(c) Perfect Hashing

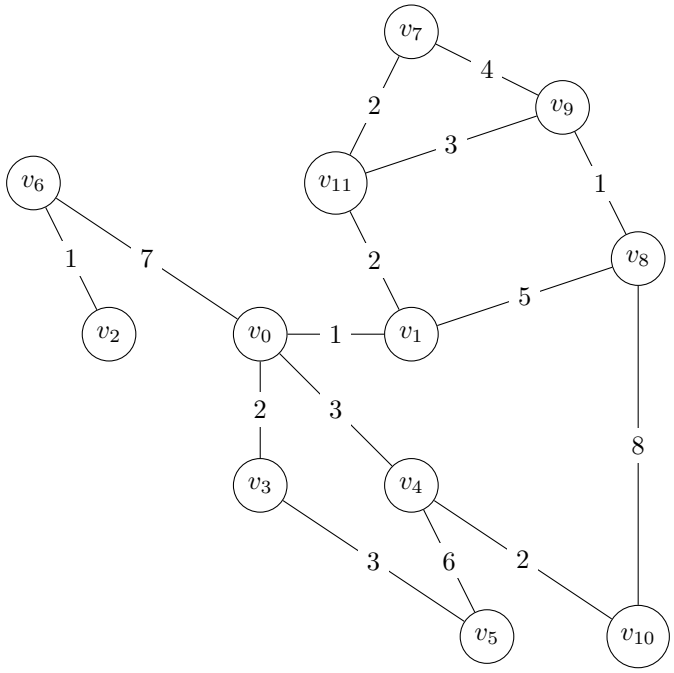
(d) Maximum Path Height

(e) Primary Key

Question 8 Construct a *Red-Black Tree* by inserting the following values in-order. Show all intermediate steps. Indicate a red node with a dashed circle, and a black node with a solid circle.

`10, 5, 15, 0, 2, 12, 20, 13, 11, 14`

Question 9 Use Kruskal's algorithm to construct a minimum spanning forest for the following graph. Yes, you may the draw graphs. Show all steps.



Question 10 Is the following function a "good" candidate for a hashing function? Why or why not? Supply evidence for your claim.

```
public class Hasher {
  public static int hash (int input, int limit) {
    int odd = input;

    if (odd % 2 == 0) odd += 1;

    input = 0;
    while (odd > 0) {
      input += odd % 10;
      odd /= 10;
    }

    return input % limit;
  }
}
```