Name (print): _____

PID: _____

COMP 410 Spring 2019

Final Exam

This exam is closed book, notes, calculators, cell phones, classmates, smart watches, wireless internet brain connections, closed everything but your own brain. You have 180 minutes to complete the exam. Do all your work on these exam pages. Please sign here pledging that the work you submit is your own:

Print your name up at the top of each page.

Signature: _____

Q1 (8%) True or False (T/F) ***For the following, "graph" means "connected graph" :***

a) _____ An undirected graph with all edge weights the same has only 1 minimum spanning tree

b) _____ The splay function in a splay tree might lengthen the longest path rather than shorten it

c) _____ If the number of nodes in a tree is odd, then that tree is a bi-partite graph

d) _____ Every graph that is a tree has no Euler path

e) _____ No complete graph is planar

f) _____ Every non-planar graph has at least one spanning tree

g) _____ A hash table using probing cannot fill up... one more item can always be inserted

h) _____ A hash table using chaining cannot fill up… one more item can always be inserted

Q2 (2%): Consider this code fragment for function **funOne**:

```
public static long funOne(int N) {
  long sum = 0;                                  answer: _____________
  for (int i=1; i<=N; i*=2) {
    for (int k=0; k<5; k++) { sum = sum + i*k; }
  }
  return sum;
}
```

For N a positive integer, what is a good "Big Oh" worst case execution time of function **funOne** ?

Q3 (2%): Consider this code fragment for function **bar:**

```
public static long bar(long K) {
  if (K == 2) return 2;                          answer: _____________
  return bar( bar(K-1) );
}
```

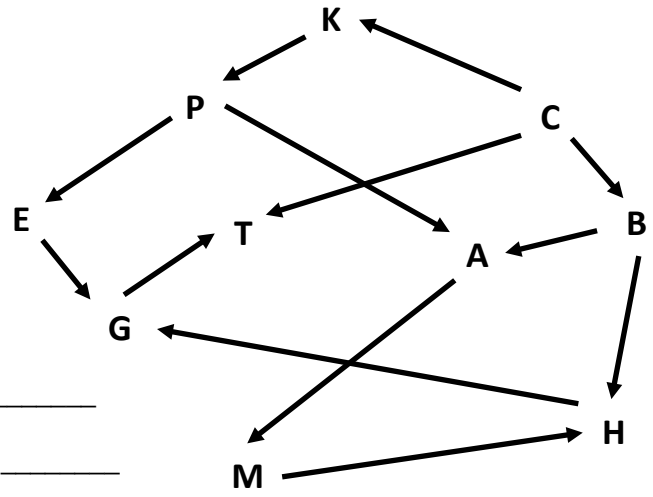
For K a positive integer >= 2, what is a good "Big Oh" worst case execution time of function **bar** ?

Name (print): _____

For **Q4** and **Q5,** consider the di-graph to the right:

Also consider these sequences of nodes

a) C B K T A H P M G E
b) C B H G T P E A M K
c) K P C E T A B G M H
d) T G E P K A M H B C
e) C K P E G T A M H B
f) none of these



Q4 (2%): Which sequence is a depth-first search? _____

Q5 (2%): Which sequence is a breadth-first search? _____

Q6 (1%): Which of the following is the best high-level description of the blockchain (such as used to implement Bitcoin)?

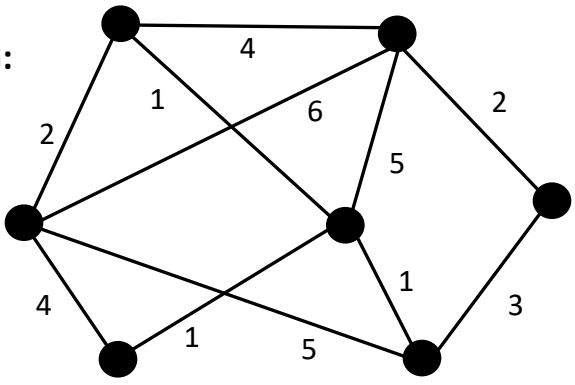
a) Skip list that uses hashing instead of coin flips
b) Balanced binary search tree with hashes as node values
c) Hash map of data blocks that uses chaining for collisions
d) Linked-implementation stack with hash values as links
e) Doubly linked list with hash values as links
f) Minimum binary heap of hashes as priorities

Answer: _____

Q7 (6%)

True or False: Consider the graph **G** to the right, then answer T or F (true or false) for each of these:

G:



a) _____ G is a complete graph

b) _____ G has more than 5 spanning trees

c) _____ G has more than one minimum spanning tree

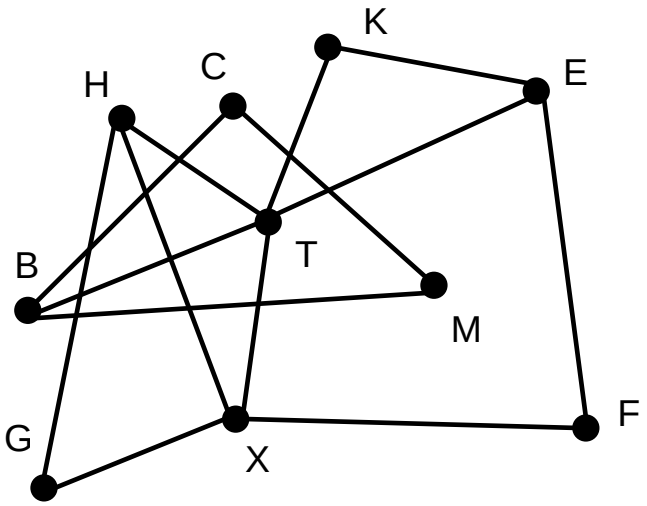
d) _____ G has an Euler circuit

e) _____ G has an Euler path (that cannot be made a circuit)

f) _____ G is not a planar graph

Name (print): _____

Q8 (4%) Consider this undirected graph (at right)



a) Is there a Hamiltonian path? _____

b) If so, show one:

c) If not, why not?

d) Is there a Hamiltonian circuit? _____

e) If so show one:

f) If not, why not?

Q9 (12%) For this question, you will select from this list of items:

```
(A) recursion                          (L) merge sort
(B) no efficient solution is known     (M) SHA-256
(C) insertion sort                     (N) is efficiently solvable
(D) heap sort                          (O) dynamic memory in Java
(E) can be solved in constant time     (P) bucket sort
(F) nonce                              (Q) skip list
(G) sparse graph                       (R) solution is impossible
(H) no efficient solution is possible  (S) selection sort
(I) trie                               (T) dense graph
(J) Hash table with chaining           (U) quick sort
(K) average list length                (V) traveling salesman problem
```

For each of the following definitions, select the item above that **best matches** (put the corresponding letter in the blank); *any item above may be used more than once:*

i) _____ lambda for a chaining hash table

ii) _____ finding a minimum spanning tree for a graph

iii) _____ used for type ahead suggestions in apps like google search

iv) _____ hash function used for cryptography and block-chain mining

v) _____ the halting problem

vi) _____ sort that has near linear best case time complexity

vii) _____ run time heap, and call stack

viii) _____ find a Hamiltonian path in a graph

ix) _____ complete graph with 8 vertices

x) _____ "fudge factor" integer used to create a valid data fingerprint for a blockchain block

xi) _____ probabilistic alternative to balanced BST

xii) _____ graph formed with a node for every person in the US and an edge between every pair of people that know each other

Name (print): _____

Q10 (2%) Fill in the table with **worst-case** time complexity for these operations

operation	Print all possible shuffles of N cards	Print all subsets of a set of size N
Big Oh	a)	b)

Q11 (8%) Choose from this list to answer below (N items stored in each data structure)

- **A.** Basic linked list (not sorted)
- **B.** BST tree (with no balancing being done)
- **C.** Hash table
- **D.** AVL tree
- **E.** Splay tree
- **F.** Skip list
- **G.** None of these (you may use this multiple times if you wish)

Data Structure	*Worst case find time complexity*	*Average case find time complexity*
i) _____	**O(N)**	**O(1)**
ii) _____	**O(N)**	**O(log N)**
iii) _____	**amort O(log N)**	**O(log N)**
iv) _____	**O(log N)**	**O(1)**
v) _____	**almost impossible O(N)**	**O(log N)**
vi) _____	**O(N)**	**O(N/2)**
vii) _____	**O(1)**	**O(1)**
viii) _____	**O(log N)**	**O(log N)**

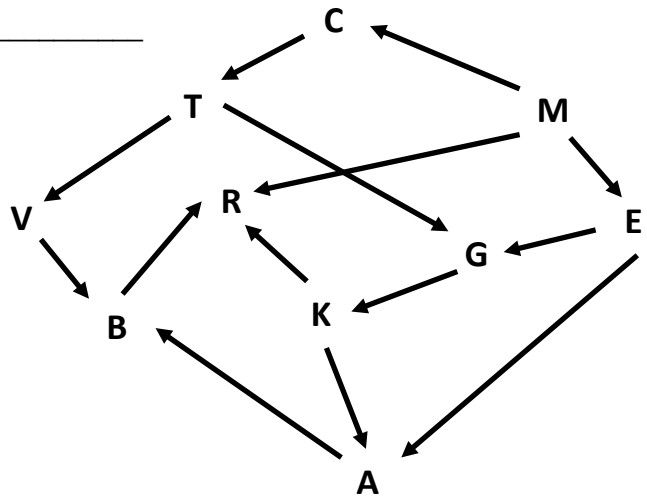
Name (print): _____

Q12 (6%): Topological sort

For the graph shown following, give a topological sort. **When you have a choice for next node, choose the smallest alphabetically**. Show your work for possible partial credit.

Answer: _____

Work:



Q13 (6%) True or False (T/F):

a) _____ No undirected graph can have only one node with odd degree

b) _____ No polynomial time algorithm can exist to find Hamiltonian paths in a graph

c) _____ In a balanced AVL tree, the shortest path and the longest path may differ by more than 1

d) _____ There is no known linear time algorithm for finding Euler paths in a graph

e) _____ The SHA-256 hash function guarantees that any 2 unique strings of text will produce 2 different values when hashed

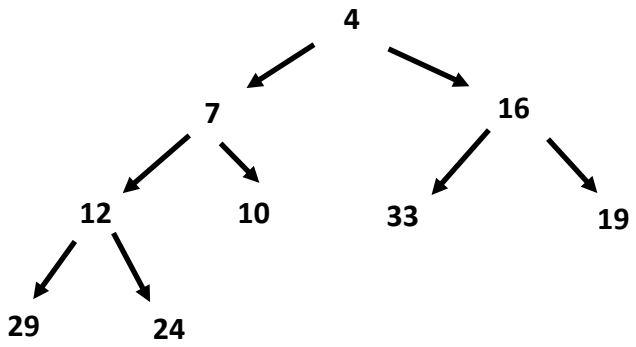
f) _____ If a undirected graph with weighted edges has exactly one minimum spanning tree then two edges might have the same weight

Q14 (3%) Consider the structure represented to the right.

a) (T/F) _____ This could be a min binary heap

b) (T/F) _____ This could be a splay tree

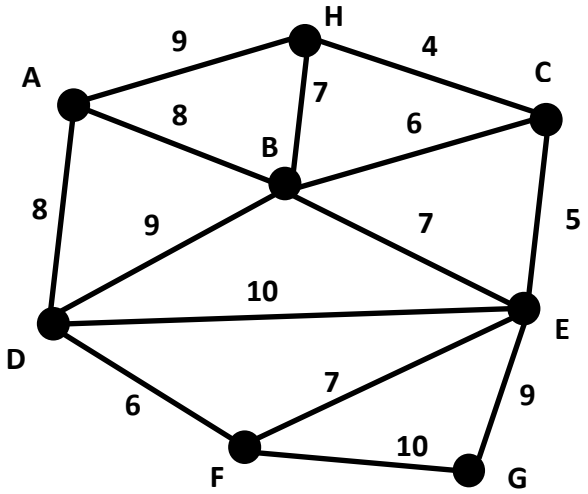
c) (T/F) _____ This could be a (balanced) AVL tree



Name (print): _____

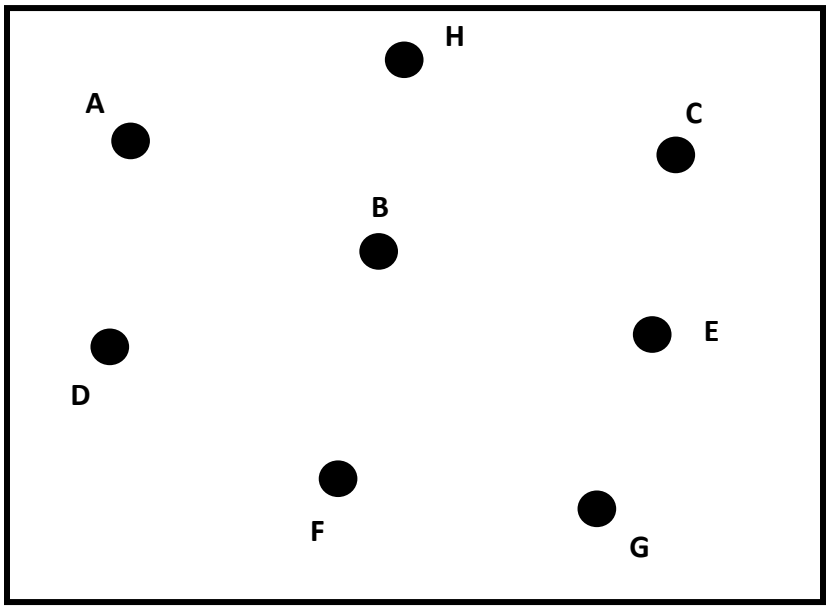
Q15 (6%) Consider the undirected graph below. Using Kruskal's algorithm, find a Minimum Spanning Tree. *Show your work for full credit.*

Draw your final MST in the box at the bottom of the page. Just add edges to the nodes that are there. Mark the weights on the edges. Write the min length in the answer space given..



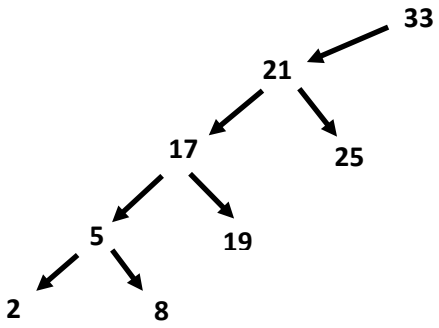
MST:

Min length: _____



Name (print): _____

Q16 (3%) Consider the structure represented to the right



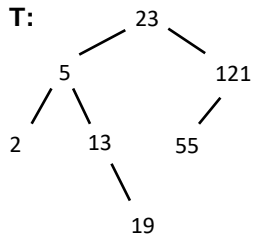
a) (T/F) _____ This could be a max binary heap

a) (T/F) _____ This could be a basic BST (no balancing)

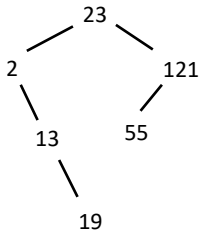
b) (T/F) _____ This could be a splay tree

Q17 (3%) Consider the AVL tree T shown below.
Which of the trees (a,b,c) below is the one that results if a **delete(5)** is done on **T**? _____

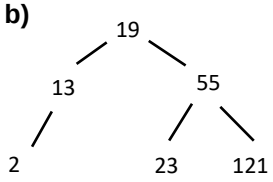
T:



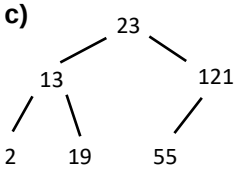
a)



b)

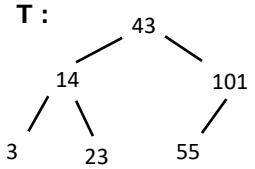


c)

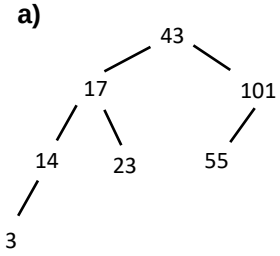


Q18 (3%) Consider the AVL tree **T** shown below.
Which of the trees (a,b,c) below is the one that results if an **insert(17)** is done on **T** ? _____

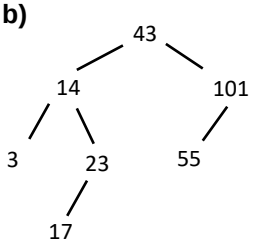
T :



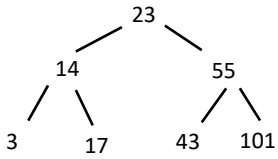
a)



b)

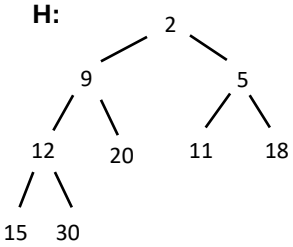


c)

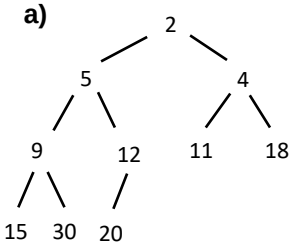


Q19 (3%) Consider the min binary heap **H** shown below.
Which of the heaps (a,b,c) below is the one that results if a **add(4)** is done on **H** ? _____

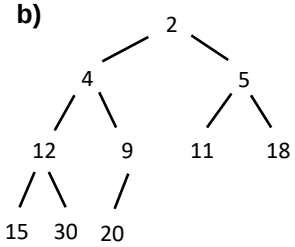
H:



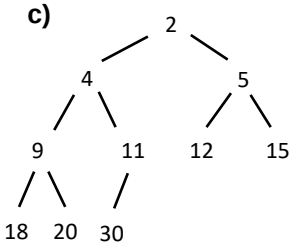
a)



b)

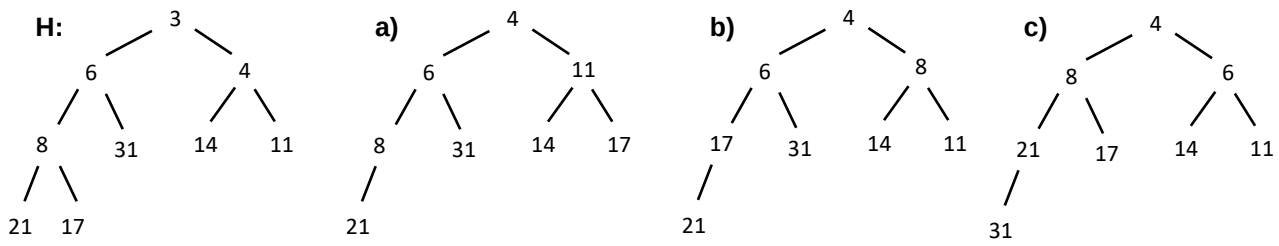


c)

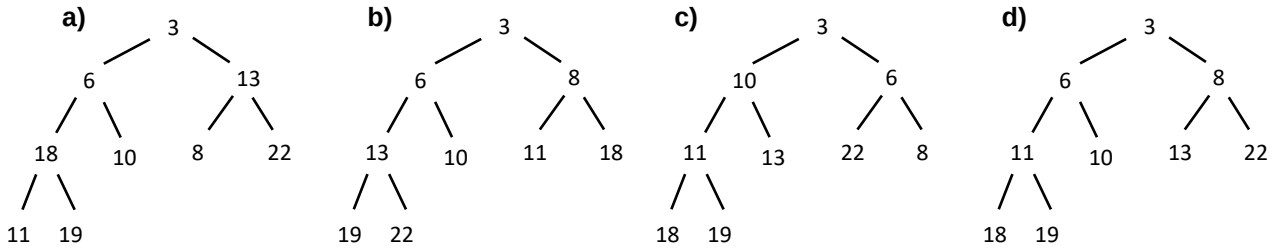


Name (print): _____

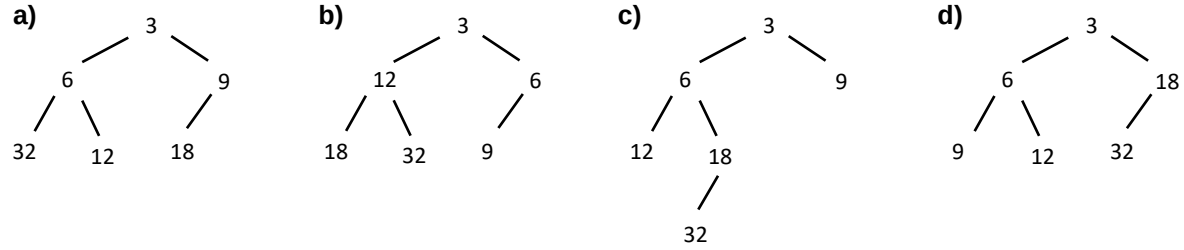
Q20 (3%) Consider the min binary heap **H** shown below.
Which of the heaps (a,b,c) below is the one that results if a **delMin()** is done on **H** ? _____



Q21 (3%) Consider this sequence **S** of values: **13, 18, 22, 6, 10, 8, 3, 11, 19**
Which heap below (a,b,c,d) results when the **"magic" build** is done on **S** ? _____



Q22 (3%) Consider this sequence S of values: **18, 32, 6, 12, 3, 9**
Which heap below (a,b,c,d) results when the elements in **S** are added
to an initially empty heap one at a time in the left-to-right order shown? _____



Name (print): _____

Q23 (9%) We are going to sort N items given to us in an array. We will return the items in the same array in sorted order. Fill in the table below by choosing a letter from the list that names a sort with the best fit for the characteristics given on each row.

- **A.** Bubble sort
- **B.** Selection sort
- **C.** BST sort (no balancing)
- **D.** AVL tree sort
- **E.** Skip list sort
- **F.** Heap sort
- **G.** Merge sort
- **H.** Quick sort
- **I.** None of these (you may use only once, like the others)

Sort type	***Worst case time to sort***	***Avg case time to sort***	***In place (in the array)***	***Stable (easily made stable)***
i) _____	**O(N^2)**	**O(N log N)**	**NO**	**NO**
ii) _____	**O(N^2)**	**O(N^2)**	**YES**	**YES**
iii) _____	**O(N^2)**	**O(N^2)**	**YES**	**NO**
iv) _____	**Very unlikely O(N^2)**	**O(N log N)**	**YES**	**NO**
v) _____	**O(N log N)**	**O(N log N)**	**NO**	**YES**
vi) _____	**O(N log N)**	**O(N log N)**	**YES**	**NO**
vii) _____	**O(N log N)**	**O(N log N)**	**YES**	**YES**
viii) _____	**O(N log N)**	**O(N log N)**	**NO**	**NO**
ix) _____	**Nearly impossible O(N^2)**	**O(N log N)**	**NO**	**YES**