Computer Science 260 Practice for the Final
Out of XX possible marks NAME:_____

1. (8 marks) Circle either **T** (True) or **F** (False), whichever is a more accurate assessment of the following statements:

T	**F**	When hashing with chaining, the number of slots must be at least as great as the number of keys to be stored.
T	**F**	To rebalance and AVL tree after an insertion, we always rebalance at the root first.
T	**F**	If $f(n) \in \mathbf{O}(g(n))$ then $\frac{1}{f(n)} \in \Omega(\frac{1}{g(n)})$
T	**F**	A good hash function guarantees that the keys will distribute uniformly among the slots.
T	**F**	If $2^{f(n)} \in \mathbf{O}(2^{g(n)})$ then $f(n) \in \mathbf{O}(g(n))$
T	**F**	If $f(n) \in O(g(n))$, then $\lg(f(n)) \in \mathbf{O}(\lg(g(n))$
T	**F**	All the Minimum Spanning Tree (MST) algorithms we studied are examples of greedy algorithms.

2. Consider the following MST algorithm:

```
Algorithm MST (weighted Graph G=(V,E), weight function w)
// precondition: G is a connected graph with m edges with real-values edge weights
// postcondition: Tree is a Minimum Spanning Tree of G

sort the edges of E in NON-ASCENDING order of weight
```

(so that $e_1, e_2, \ldots, e_m$ is such that that $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_m)$)

```
ecounter = m
Tree = G
i = 1
while (ecounter > |V|-1) do
    if removing e_i from Tree does not disconnect Tree
         remove e_i from Tree
    ecounter--
    i++
return Tree
```

   (a) (3 marks) Is Algorithm MST correct? Yes or no. If not, modify it slightly so it works.
   (b) (3 marks) Suppose you can check the connectivity of a graph $G = (V, E)$ in $\mathbf{O}(m)$ time, where $m = |E|$. What is the running time of the above algorithm?

3. Analyze the running time of the following procedure:

```
Funny(A[0..n-1])

   if  n==0 then return 1
   if  n==1 then return A[0]

   do O(\sqrt{n}) amount of work that does not affect A;

   return (Funny(A[0..\floor{\frac{n}{3}}] + Funny(A[\floor{\frac{n}{3}}+1..n])
```

4. (4 marks) Place the following functions in a 'Big Theta' hierarchy – place the slow-growing ('good' running time) functions at the bottom, etc. Two functions should be on the same level of the hierarchy if they are related by Big Theta.

   n^3
   $n \log n$
   $n^{\frac{3}{4}}$
   $n^2 - n - 1$
   $100n^2 + n \log n$
   $\frac{n!}{2^n}$
   $(\frac{n}{2})!$
   $n \lg^1 000n$

5. (4 marks) Either prove that $\lg^2 n$ is in $\mathbf{O}(\lg n)$ or prove that it is not.

6. (4 marks) Prove using whichever method you prefer that $3n^2 + 9n^{1.9}\lg^2 n \in \Theta(n^2)$.

7. (4 marks) Prove using whichever method you prefer that $\frac{n^6}{\lg^2 n} \in \mathbf{O}(\frac{n^6}{\lg n})$.

8. (4 marks) Prove using whichever method you prefer that $14n^2 \log n + 9n \lg n + 100 \in \Theta(n^2 \log n)$.

9. (4 marks) Prove using whichever method you prefer that $n^{5.5} \notin O(n^{5.4})$

10. (4 marks) Prove the Product Rule of Big-Oh.

11. (4 marks) Prove the Transitivity Rule of Big-Oh.

12. (4 marks) Prove the Transitivity of Θ.

13. (3 marks) Write the recursive version of Quick-Union implementation (i.e., the forest implementation) of find(x) that uses path compression.

14. (4 marks) Suppose after a sequence of Disjoint Set operations, the forest data structure is as shown below. Show the state of the forest after a call to union(a, b). Then show the forest after a call to find(x). Assume path compression and union-by-rank are used. Show the rank of the roots of the trees in the resulting forest.

15. (4 marks) Run Dijkstra's algorithm on the following graph. In each column, show the values of the current 'label' on each vertex that is in the heap. On the head of each column, write the vertex selected.

16. (6 marks) Use Dynamic Programming to find the longest common subsequence (not necessarily contiguous) in the following two strings, by filling the table below.

	B	A	C	D	B	C	A	D
A								
B								
A								
C								
C								
A								
D								
B								

17. (6 marks) Analyze the following algorithm, and give its running time. Assume that the divisions by 2 or 4 are done using floors or ceilings in a sensible way.

```
Algorithm Splog (A[1..m][1..m], m)
// The input is an m x m array
   if m>3 {
      Splog(A[1..m/2] [1..m/2], m/2)
      Splog(A[m/2..m] [m/2..m], m/2)
      Splog(A[1..m/2] [m/2..m], m/2)
      Splog(A[m/2..m] [1..m/2], m/2)
      Splog(A[m/4..3m/4][m/4..3m/4], m/2)
      Sew-Up(A,m)
   }
```

The procedure Sew-Up takes $m^2 \lg m$ time.

Note: the input is of size m^2. It may help to let $n = m^2$ and determine your running time as a function of n.

18. Give the running times for algorithms whose running time is represented by the following recurrences, or indicate that the Master Theorem is silent on the case:

(a) (2 marks) $T(n) = 9T(n/3) + n^2 \log n$

(b) (2 marks) $T(n) = 15T(n/4) + n^2 \log n$

(c) (2 marks) $T(n) = 4T(n/4) + 2n$

(d) (2 marks) $T(n) = 3T(n/4) + \sqrt{n}$

19. (5 marks) Perform a BST insert of 12 into the following BST tree.

20. (5 marks) Execute an AVL-Tree insert of these keys into an initially empty AVL tree: 20, 10, 30, 25, 40, 50, 15, 22, 23. Show the resulting AVL tree. Then execute a Delete(50), and before rebalancing, identify teh lowest node that is out of balance by circling it. Then do the appropriate rotations to restore balance, and show the resulting AVL tree.

21. (4 marks) Define 'clustering' and describe a strategy for avoiding it when using closed hashing.

22. (3 marks) Perform a removeMin on the following heap.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

23. Consider the hash function $h(c_1c_2 \ldots c_n) = \Sigma_{i=1}^{n} \text{ord}(c_i) \text{mod} m$.

    (a) (1 mark) What is does 'm' refer to in the above hash function?

    (b) (6 marks) Insert the following keys into the open hash table below, using closed hashing with double hashing to handle collisions. The secondary hash function is "string length + number of vowels" mod m. AN MB AR WX DE LN GI UF

    

    (c) (2 marks) Why is it a bad idea if the value of m were 12? What kind of behaviour might it lead to in insertion and in deletion?

    (d) (3 marks) Write the Find(key) algorithm for the above table. What happens if some elements have been deleted?

24. (6 marks) Give a simple, efficient, recursive, Dynamic Programming algorithm to perform exponentiation. Return the expected value.

```
Algorithm Expon( $a$, $n$ )
/* input: $a$ is a floating point number, $n$ is an non-negative integer
/* output: h$a^n$ */
```

25. The Master Theorem for finding closed form expressions for recurrence relations is expressed as follows. For a function $T(n)$ defined on positive integers, where $T(n) = aT(\frac{n}{b}) + f(n)$ and $f(n)$ is a positive-valued function, and constants a and b are such that $a \geq 1$ and $b > 1$, then:
1. **If** $f(n)$ is $O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, **then** $T(n)$ is $\Theta(n^{\log_b a})$.

2. **If** $f(n)$ is $\Theta(n^{\log_b a})$ **then** $T(n)$ is $\Theta(n^{\log_b a} \log n)$

3. **If** $f(n)$ is $\Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and there exists some $c, 0 < c < 1$ such that $af(\frac{n}{b}) < cf(n)$ when n is sufficiently large, **then** $T(n)$ is $\Theta(f(n))$.

   (a) (3 marks) $T(n) = 8T(\frac{n}{2}) + n^3$
   (b) (3 marks) $T(n) = 4T(\frac{n}{2}) + n^3 \lg n$
   (c) (3 marks) $T(n) = 4T(\frac{n}{2}) + \frac{n^2}{\lg n}$
   (d) (3 marks) $T(n) = 2T(\frac{n}{2}) + \frac{n^2}{\lg n}$

26. (5 marks) Prove the following is true: If $G = (V, E)$ is a graph such that every cut in G has exactly one light edge, then G has exactly one Minimum Spanning Tree. (Recall that an edge is light if it is a least weight edge for the cut; and a cut is a partition $(S, V - S)$ of the vertices of G, and an edge is in a cut if it has one end in S and the other in $V - S$.)

27. (5 marks) On the following graph, run Prim's algorithm to find the MST. Number each edge that gets added, and put that number in the parentheses for that edge, i.e. "1" for the first edge added, etc. Start at vertex s. There are 17 edges and nine vertices.

28. (5 marks) Do the same using Kruskal's algorithm.

29. (8 marks) Run Dijkstra's Algorithm on the following graph and vertex s. At each iteration, put the added vertex on top of the column and the new key values for all affected vertices in the column, as we did in class.

30. Give a set of coin denominations for which the greedy algorithm does <u>not</u> provide the optimal (minimum) number of coins. Prove your claim by giving an amount C, and the change made by the greedy algorithm, and a way to make the change that uses fewer coins.

31. Construct and fill the LCS ("Longest Common Subsequence") dynamic programming table for the sequences

A B B A C B D A C C D B A B C C B D C

32. (10 marks) Circle either **T** (True) or **F** (False), whichever is a more accurate assessment of the following statements:

T **F** $f(n)$ is $O(g(n))$ if and only if $\frac{1}{f(n)}$ is $\Omega(\frac{1}{g(n)})$.

T **F** Dijkstra's algorithm, adapted for graphs with adjacency matrix representation, runs on dense graphs in time $O(|V|^2)$.

T **F** $f(n)$ is $o(g(n))$ if for any c there is an N such that $f(n) < c(n) \ \ \forall n > N$.

T **F** When hashing, the number of slots must be at least as great as the number of keys to be stored.

T **F** The Master Theorem solves all recurrences of the form $T(n) = aT(\frac{n}{b}) + f(n)$ whenever $a \geq 1, b > 1$, and $f(n)$ is a positive-valued function

T **F** An AVL tree is a balanced, heap-ordered Binary Tree.

T **F** In an array-based heap where the array is indexed from 0, a node that is not the root and has index i has parent with index $i/2 + 1$

33. (6 marks) Order the following functions in ascending order of growth rate. That is, write the number "1" beside the **slowest** growing function, "2" beside the next-slowest, and so on. If two functions have the same growth rate asymptotically, give them the same number.

_____ $1.4^n \log n$.
_____ $\log n + \frac{n+1}{n}$.
_____ $15n^{\frac{1}{3}}$.
_____ $1.3^n \log_2 n$.
_____ $\log_6^2 n - \log_5 n$.
_____ $2^{\sqrt{\log n}}$.

34. (2 marks) State the Product Rule for Big Oh.

35. (5 marks) Prove using only the Facts of Big-Oh that $\frac{1}{3} n \log n + \frac{5}{\log n}$ is $O(n^2 \log \log n)$

36. (5 marks) Prove using only the definition of Big-Oh that $6n^{10} + 4n^2$ is $O(n^{10} \log n)$.

37. (5 marks) Prove using the Facts and/or the definition of Big-Oh that $5n^3 / \log^3 n + 100n \log n$ is $O(n^4 / \log^4 n)$. If using the Facts, refer to them explicitly.

38. Give the best Big Theta (Θ) running time of the following code fragments, as a function of n. You do not have to justify your answer if it is correct. If your answer is not correct, you will not get part marks unless you very clearly show your logic and your logic is partly correct.

(a) (5 marks)

```
for (i=1; i <= n; i = i*2)
    for (j=1; j<=i; j++)
        sum = sum++;
```

(b) (5 marks)

```
for (i=1; i<=n; i++)
    for(j=i; j<=n; j++)
        for (k=i; k<=j; k++)
            sum++;
```

39. Describe how you would efficiently find the record with minimum key value for each of the following Data Structures, and give the asymptotic running time for the algorithm. Your description can be a single line (or more, if necessary) in English, or pseudocode. CLARITY is important! Assume that the insert algorithms are the standard ones, and cannot be changed.

    (a) (4 marks) AVL tree of n nodes.

    (b) (4 marks) Heap of n nodes.

    (c) (4 marks) Hash table, table size m, number of keys in it is n where $n \leq m$, and the Universe of key values is 2^{10}, which is much bigger than m; and Open Addressing is used.

40. (5 marks) Suppose an element is inserted into an AVL tree and its position, before rebalancing, is as shown below, with relative heights as shown. Perform the necessary rebalancing and give the AVL tree, with nodes and subtrees labelled, that results.

41. (5 marks) Put the following values into an AVL tree, in the order given.

    15, 4, 2, 9, 7, 6, 18, 19.

42. (5 marks) Use the efficient MakeHeap algorithm on the following array as a min-heap. Show each switch.

43. (2 marks) Which graph representation scheme, of the two we studied in class, is best for implementing Dijkstra's Shortest Paths algorithm (the one we implemented in the lab)? Under what circumstances is it better?

(a) (2 marks) What best describes the ADT that hashing is designed to implement: Dictionary, Priority Queue, Graph, Heap, Sparse Table, Ink Blot, Binary Counter.

(b) (4 marks) Describe in general terms the strategy for handling collisions utilized by Open Hashing (with chaining).

(c) (4 marks) Describe Open Hashing. Under what circumstances is it preferred to Closed Hashing (all keys in the table)?

(d) (4 marks) Write the pseudocode for ChainedHashDelete($T, x,$), where T is a hash table, x is the key value of the item to be deleted, and collisions are handled by chaining.

(e) (5 marks) Under the assumption of Simple Uniform Hashing, if x and y are two keys selected from a set of keys K, where K is a subset of the Universe of keys U, and the hash table is of size m, determine the following, in terms of m, $|K|$ and $|U|$.

i. the probability that x and y hash to the same slot

ii. if 100 keys have already been inserted, and Open Addressing is used, the probability that the first probe when hashing x has a collision

iii. if chaining is used, what is the average chain length after the insertion of all $|K|$ keys?

iv. if chaining is used, what is the average number of probes for a successfule search?

v. In the following situation, if Open Addressing with and quadratic probing is used, where $h(k,i) = (h(k) - 2i + i^2) \text{mod} 11$, show the result of inserting A, B, C, and D, in that order, if $h(A) = 4, h(B) = 4, h(C) = 7, h(D) = 4$.

Priority Queues:

(a) List the operations on a Priority Queue.

(b) State briefly in English how you'd implement these operation if the data structure was an unsorted array.

(c) ...a sorted array.

(d) What other implementation did we discuss in class?

44. Suppose you have a B-tree, with $t = 2$ with the following nodes:
root = DIP
AB EGH LNO RTV
Suppose an BTreeInsert(F) is executed. What B-tree results?
Then suppose a BTreeInsert(Q) is executed. What B-tree results?

45. Solve the following recurrence using the Master Theorem. (4 marks) $T(n) = 5T(\frac{n}{5}) + 5n \log n$

Definition: Big Oh $f(n) \in \mathbf{O}(g(n)) \iff \exists$ positive constants c, n_0 such that$f(n) \leq c \cdot g(n)$, $\forall n \geq n_0$.

Big O Rules (Facts):

(a) Transitivity of Big O: $f(n) \in \mathbf{O}g(n)$ and $g(n) \in \mathbf{O}(h(n)) \Rightarrow f(n) \in \mathbf{O}(h(n))$.

(b) Strange But True Log Domination Rule: $(\log n)^r \in \mathbf{O}(n^s)$ for all constants r and s both not equal to 0.

(c) Polynomial Rule: $p(n) \in \mathbf{O}(q(n))$, whenever $p(n)$ is a polynomial in n, of degree k, and $q(n)$ is a polynomial in n, of degree t, where k and t are constants and $k \leq t$.

(d) Product Rule: $f_1(n) \in \mathbf{O}g_1(n)$ and $f_2(n) \in \mathbf{O}(g_2(n)) \Rightarrow f_1(n) \cdot f_2(n) \in \mathbf{O}(g_1(n) \cdot g_2(n))$.

(e) Removal of Constant Factors Rule: $f(n) \in \mathbf{O}(c \cdot f(n))$ for all constants $c > 0$.

(f) Log Base is Irrelevant Rule: $\log_a n \in \mathbf{O}(\log_b n)$ for all constants $a, b > 0$ and $a, b \neq 1$.

(g) Reciprocal Rule: $f(n) \in \mathbf{O}((g)n)$, then $\frac{1}{g(n)} \in \mathbf{O}(\frac{1}{f(n)})$.

(h) Sum Rule: If $f_1(n) \in \mathbf{O}((g)n)$ and $f_2(n) \in \mathbf{O}(g(n))$ then $f_1(n) + f_2(n) \in \mathbf{O}(g(n))$.

(i) Less-Than Rule: if $f(n) \leq g(n)$ for all n greater than some $n_0 > 0$ then $f(n) \in \mathbf{O}(g(n))$.

Log facts

(a) $\lg n \leq n \forall n \geq 1$, and $\log n \leq n \forall n \geq 1$.

(b) $\log_a^d n$ is, by definition, $(\log_a n)^d$

(c) $2^{\lg n} = n$

(d) $\log_a n^b = b \log_a n$

(e) $\log_b a = \frac{\log_c a}{\log_c b}$

(f) $\log_b a = \frac{1}{\log_a b}$

(g) $a^{\log_b c} = c^{\log_b a}$

(h) $\log_c(a * b) = \log_c a + \log_c b$

(i) $\log_4 5 = 1.161$

(j) $\log_5 4 = 0.861$

(k) $\log_4 3 = 0.792$

(l) $\log_3 4 = 1.262$