CIS507: Design & Analysis of Algorithms
Final, Spring 2012

Duration: 100 minutes
Total weight: 30%

Student Name: –

Student ID: –

Question	Points Obtained	Points Possible
1		2
2		2
3		6
4		10
5		4
6		2
7		4
Total		30
8 (bonus)		2
Grand Total		32

Cheat Sheet: Master Method

For $T(n) = aT(n/b) + f(n)$, with $a \geq 1$, $b \geq 1$, compare $f(n)$ with $n^{\log_b a}$.

Case	Condition, for $\epsilon > 0$	Solution
1	$f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$	$T(n) = \Theta(n^{\log_b a})$ Number of leafs dominates
2	$f(n) = \Theta(n^{\log_b a})$	$T(n) = \Theta(n^{\log_b a} \lg n)$ All rows have same asymptotic sum
3	$f(n) = \Omega(n^{\log_b a + \epsilon})$	$T(n) = \Theta(f(n))$ provided that $af(n/b) \leq cf(n)$ for some $c < 1$
2 (general)	$f(n) = \Theta(n^{\log_b a} \lg^k n)$ or some constant $k \geq 0$	$T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$ They grow at 'similar' rate

1 Trace Quicksort (2 points)

Draw the recursion tree generated by Quicksort on this array:

$$A[1, \ldots, 8] = \langle 33, 55, 11, 88, 77, 22, 66, 44 \rangle$$

Label each node with the contents of the sub-array being sorted. So, the root should be labelled with the original contents of A, its two children will be the sub-arrays to be sorted recursively, etc.

Note: Make sure you move things around correctly as you do the partition. To help you, here is the partition sub-routine, which rearranges sub-array $A[p, \ldots, r]$ with pivot r. Note that the *last* element is chosen as the pivot.

```
PARTITION(A, p, r)
  x = A[r]
  i = p - 1
  for j = p to r - 1
      if A[j] ≤ x
          i = i + 1
          exchange A[i] with A[j]
  exchange A[i + 1] with A[r]
  return i + 1
```

2 Hashing (2 point)

Suppose we use simple uniform hashing, and resolve collisions by chaining. Suppose you insert three keys into a hash table with $m = 10$ slots. What is the probability that slots 0 and 1 empty?

3 Multiple Choice with One Correct (6 points)

For each of the following, circle the *single* correct answer.

1. **(1 point)** Recurrence $T(n) = 3T(n/3) + n$ has solution $T(n) =$
   (a) $\Theta(n)$ (b) $\Theta(n \lg n)$ (c) $\Theta(\lg n)$ (d) Unknown

2. **(1 point)** Recurrence $T(n) = 11T(n/7) + n^3$ has solution $T(n) =$
   (a) $\Theta(n)$ (b) $\Theta(n \lg n)$ (c) $\Theta(n^3)$ (d) Unknown

3. **(1 point)** You are given an adjacency-list representation of a directed graph with n vertices and m edges. Let k denote the maximum in-degree of a vertex. Each vertex maintains an array of its outgoing edges (but *not* its incoming edges). How long does it take, in the worst case, to compute the in-degree of a given vertex? (Recall that the in-degree of a vertex is the number of edges that enter it.)
   (a) $\Theta(k)$ (b) $\Theta(m)$ (c) $\Theta(n)$
   (d) $\Theta(k + m)$ (e) $\Theta(n + m)$ (f) $\Theta(k * m)$

4. **(1 point)** The $\leq_p$ relation (polynomial-time reducible) is a transitive relation.
   (a) Yes (b) No (c) For some classes of problems (d) Unknown

5. **(1 point)** A directed graph is called *strongly connected* if there is a path from each vertex in the graph to every other vertex. The *strongly connected components* of a directed graph G are its maximal strongly connected subgraphs. On adding one extra edge to a directed graph G, the number of strongly connected components:
   (a) might remain the same.
   (b) always decreases.
   (c) always increases.
   (d) always changes.

6. **(1 point)** We can prove that problem X is NP-Hard by:
   (a) Showing a polynomial time reduction from 3-SAT to X
   (b) Showing a polynomial time reduction from X to 3-SAT
   (c) Either of the above
   (d) None of the above

4 True or False (10 points)

1. **(1 point)** If $T_1(n) = \mathcal{O}(f(n))$ and $T_2(n) = \mathcal{O}(f(n))$, then $T_1(n)+T_2(n) = \mathcal{O}(f(n))$.

2. **(1 point)** If $T_1(n) = \mathcal{O}(f(n))$ and $T_2(n) = \mathcal{O}(f(n))$, then $\frac{T_1(n)}{T_2(n)} = \mathcal{O}(1)$.

3. **(1 point)** If $T_1(n) = \mathcal{O}(f(n))$ and $T_2(n) = \mathcal{O}(f(n))$, then $T_1(n) = \mathcal{O}(T_2(n))$.

4. **(1 point)** Using a comparison sorting algorithm, we can sort any 7 numbers with up to 12 comparisons.

5. **(1 point)** Counting sort is *not* a comparison sorting algorithm.

6. **(1 point)** Running merge sort on an array of size n which is already correctly sorted takes $\mathcal{O}(n)$ time.

7. **(1 point)** If a problem is $NP - complete$, then no polynomial time algorithm exists for solving it.

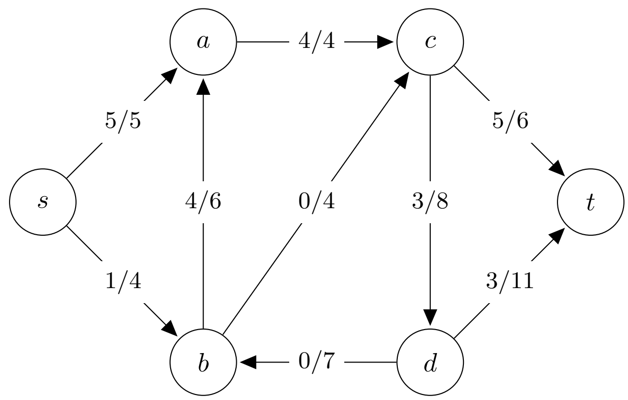
8. **(1 point)** Suppose we know that a problem X is NP-complete. If we find a polynomial-time algorithm for X, this mean that we can solve SATISFIABILITY in polynomial time.

9. **(1 point)**) Suppose we know that a problem X is in NP. If we find a polynomial-time algorithm for X, then we can solve SATISFIABILITY in polynomial time.

10. **(1 point**) Suppose we find an $\mathcal{O}(n^2)$ algorithm for SATISFIABILITY. This implies that every problem in NP can be solved in time $\mathcal{O}(n^2)$.

5 Flow (4 point)

Draw the residual graph of the following flow graph:



6 Amortized Analysis (2 point)

Consider a linked list that has the following operations defined on it:

Operation	Description	Cost
$AddLast(x)$	Adds the element x to the end of the list	1
$RemoveFourths()$	Removes every fourth element in the list i.e. removes the first, fifth, ninth, etc., elements of the list.	Equals the number of elements in the list

1. Assume we perform n operations on the list. What is the worst case (asymptotic) run time of a call to $RemoveFourths()$?

2. Using the accounting method, compute the amortized cost per operation for a sequence of these two operations (give the amounts that you will charge $AddLast()$ and $RemoveFourths()$, and show how you will use these charges to pay for the actual costs of these operations). Keep your answer brief.

7 Multiple Choice with Zero or More Correct (4 points)

For each of the following, circle all (zero or more) correct answer(s). You will lose 0.25 points per incorrect choice.

1. **(1 point)** Showing a polynomial time reduction from 3-SAT to problem X proves that X is:
   (a) P (b) NP (c) NP-Complete (d) NP-Hard

2. **(1 point)** If X is NP-Complete, this implies that X is:
   (a) NP (b) EXP (c) P (d) NP-Hard

3. **(2 point)** 3-SAT is:
   (a) P (b) NP (c) CoNP (d) NP-Hard
   (e) coNP-Hard (f) NP-Complete (g) CoNP-Complete (h) EXP

8 Bonus Question (2 points)

0.5 point per choice.

Consider graphs that are undirected, unweighted, and connected. The *diameter* of a graph is the maximum, over all choices of vertices s and t, of the shortest-path distance between s and t. Next, for a vertex s, let $l(s)$ denote the maximum, over all vertices t, of the shortest-path distance between s and t. The *radius* of a graph is the minimum of $l(s)$ over all choices of the vertex s. Which of the following inequalities always hold (i.e., in every undirected connected graph) for the radius r and the diamter d?

(a) $r \geq d/2$ (b) $r \leq d$ (c) $r \leq d/2$ (d) $r \geq d$