Graduate Algorithms	Manuel Blum, Virginia Vassilevska, Chris Wang
Problem Set 0 (Last Modified Date: January 14, 2005)	Due: January 19, 2005

This assignment should be done individually: *no collaboration is allowed.* Also, please let us know, for each question, if you have seen the question before. You will be graded out of 100 points for this assignment. Please submit your solution in ALL of the following formats, both by email to `virgi+algo@cs.cmu.edu` and `chengwen+gradalg@cs.cmu.edu`, and as a hardcopy in class: ps, pdf and tex.

Problem 1 Find the asymptotic bound in terms of Θ notation for each $T(n)$. (State any condition you need)

1. $T(1) = 4;\ T(n) = T(\lfloor \frac{n}{2} \rfloor) + 1$
2. $T(-1) = -5;\ T(n) = T(n-1) + \pi$
3. $T(0) = 1;\ T(n) = 3T(\lfloor \frac{n}{3} \rfloor) + n$
4. $T(1) = 0;\ T(n) = 2T(\lfloor \frac{n}{2} \rfloor)$
5. $T(0) = 2;\ T(n) = T(\lfloor 2n/3 \rfloor) + T(\lfloor n/3 \rfloor) + \sqrt{n}$

Grading: 2pts each.

Problem 2 For each of the following either prove that it holds, or give a counterexample:

1. [2pts] For all positive functions f and g, $f(x) = O(g(x))$ implies $2^{f(x)} = O(2^{g(x)})$
2. [4pts] For *all* positive functions f and g, $f(x) = O(g(x))$ implies $\log f(x) = O(\log g(x))$
3. [2pts] For all real numbers x, $\lceil x \rceil - \lfloor x \rfloor = 1$
4. [2pts] For all real numbers a and b, $\min(a, b) + \max(a, b) = a + b$

Problem 3 [10pts] We have two dice. Let E_1 be the event that the 1st die roll is even, and E_2 be the event that the 1st and 2nd dice rolls sum to 10. Are E_1 and E_2 independent?

Problem 4 [10pts]

$$xy >= 4 \quad (1)$$
$$x + y <= 0 \quad (2)$$
$$x^2 + y^2 <= 8 \quad (3)$$

Can you solve for x and y? If so, please do so. If not, why not?

Problem 5 Let G be a directed graph represented using an adjacency list. So, each node $G[i]$ has a list of all nodes reachable in 1 step from i (all out-neighbors of i). Suppose each node of G also has a value: e.g., node 1 might have value \$100, node 2 might have value \$50, etc.

Give an algorithm that computes, for every node, the highest value reachable from that node, *i.e.* that you can get to by some path from that node. For instance, if it is possible to get to any node from any other node (G is "strongly-connected"), then for every node this will be the maximun value in the entire graph. Give an optimal bound for the running time of your algorithm.

```
Grading: The grade for this problem will fall into one of the below categories:

(n = number of vertices, m = number of edges)
5pts for a correct algorithm
10pts for a polynomial algorithm
15pts for O(m + n log n)
20pts for O(m + n)
25pts for O(n log n)
35pts for a proof that the algorithm you gave is an optimal deterministic algorithm.
```

Problem 6 [10pts] You have two decks of cards, A and B, each containing 52 cards, four cards (for each of the four suits Spades, Hearts, Diamonds, Clubs) of each of the 13 ranks (Ace, $2, \dots, 10$, Jack, Queen, King). The order of the cards in each deck is arbitrary. An evildoer takes the top card from A and the top card from B and glues them back to back. Then he takes the second card from each deck, and glues them back to back. He continues until you are left with one deck of cards, each has two faces and no back. Prove that you can find 13 cards so that all ranks of A and all ranks of B are represented.

[Hint: Hall's theorem states: Given a set A, let $N(A)$ be the set of neighbors of A. Then the bipartite graph G with bipartitions X and Y has a perfect matching iff $|N(A)| \geq |A|$ for all subsets A of X.]

Problem 7 [10pts] Let T be a tree and let $T_1, T_2, \dots, T_k$ be subtrees of T, every two of which have a vertex in common. Show that some vertex of T belongs to all of the subtrees.

Problem 8 Consider the problem of making change of n cents using the least number of coins.

(a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution. You can assume arithmetic operations take constant time.

```
Grading:
2pts for a working greedy algorithm or 5pts for an algorithm running in O(1)
5pts for the optimality proof
```

(b) [5pts] Suppose that the available coins are in the denominations $c^0, c^1, \ldots, c^k$ for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution. (You may assume c is constant but k isn't.)

(c) [5pts] Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution.

Problem 9 This problem arises from calculating success probabilities for certain role-playing games, where players roll dice in proportion to their character's abilities, and each die is either a *Success*, a *Failure* or *Neutral*, and the outcome is determined by the number of successes minus the number of failures. (For example, in one game, dice take random values from 1 to 10, with 1 being a *Failure* and 8-10 being a *Success*.) Abstractly, the problem is: there are n independent random variables, $X_1, \ldots, X_n$. Each variable is $+1$ with probability p, -1 with probability q and 0 otherwise, where $0 \leq p, q \leq 1$ and $p + q \leq 1$. (In the above example, $p = 3/10$; $q = 1/10$.) We want to calculate, given n, an array of probabilities: for all k with $-n \leq k \leq n$ compute the probability that $\sum_{i=1}^{n} X_i = k$. Give your best algorithm running in polynomial time in n. Assume addition, subtraction, multiplication and division take constant time.

```
Grading: The grade for this problem will fall into one of the below categories:
10pts for a polynomial algorithm
15pts for O(n^2)
30pts for O(n log n)
60pts for O(n)
```