Midterm Examination Problem Sheet

TIME: 04/21/2015, 14:20–16:20

This is a open-book exam. You can use any printed materials as your reference during the exam. Any electronic devices are not allowed.
Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or get negative scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.
Both English and Chinese (if suited) are allowed for answering the questions. We do not accept any other languages.

There are 5 problems in the exam, each worth 40 points—the full credit is 200 points. Some problems come with two sub-problems to help you get partial credits, making a total of 9 sub-problems. For the 9 problems, 3 of them are marked with * and are supposedly simple; 4 of them are marked with ** and are supposedly regular; 2 of them are marked with *** and are supposedly difficult.

1. C and C++

   (a) (20%, *) Give a concrete example in C/C++ to explain in your own words on how memory leak could happen.

   (b) (20%, *) Suppose we use the following class to represent the homework scores of a student in DSA class:

```
class myScore {
public:
  myScore(int n = 10){ size = n; hwScore = new int[size]; }
  ~myScore(){ delete[] hwScore; }
  string name;
  int *hwScore;
};
```

   Draw the memory layout after running the following code snippet, and explain the potential hazards/problems of the code.

```
myScore a(6);
a.name = "John"; a.hwScore[0] = 85; a.hwScore[1] = 90;
myScore b = a;
b.name = "Mary"; b.hwScore[0] = 70;
```

2. arrays and linked lists

   (a) (20%, **) Assume that you have a vector of N integers that are between 1 and M. Write down the pseudo code of an $O(N)$-time $O(M)$-space algorithm that determines whether all the integers are different from each other (that is, whether they are all distinct).

(b) (20%, *) Suppose we have a singly linked list with each node defined by the following class:

```
class node {
public:
  string data;
  node *next;
}
```

Write a C/C++ function `invert(node *head)` that can invert the list pointed to by head and return the pointer to the inverted list without doing any dynamic memory allocation. (*Hint: Note that your function should not need more than 20 lines.*)

3. stacks and queues

(a) (20%, **) Show how a stack can be used to evaluate the postfix expression

```
9 8 7 2 3 * - / - 5 2 - * 3 /
```

step by step. All the operators within the expression are binary, and you should draw the stack status after each step.

(b) (20%, **) Suppose you have a deque D that stores N elements $(d_0, d_1, \ldots, d_{N-1})$, in this order, and an initially empty queue Q. Give a pseudo-code description of a function that uses only D and Q (and no other variables or objects) such that the elements within D are reversed—that is, are stored like $(d_{N-1}, d_{N-2}, \ldots, d_0)$ in D.

4. the evil complexity

> The notation $f(n) = O(g(n))$ applies to functions $f(n)$ and $g(n)$ from $\mathbb{N}$ to $\mathbb{R}^+ \cup \{0\}$, and $f(n) = O(g(n))$ if and only if there exists $n_0 \geq 1$ and $c > 0$ such that $\forall n \geq n_0$, $f(n) \leq cg(n)$. Please only use the definition above in your proof.

(a) (20%, **) Prove or disprove the following statement: If $f(n) = O(g(n))$, then $\big(f(n)\big)^2 = O\Big(\big(g(n)\big)^2\Big)$.

(b) (20%, ***) Prove or disprove the following statement:
If $|f(n) - g(n)| = O(1)$, then $2^{f(n)} = O(2^{g(n)})$.

5. (40%, ***) Given a sorted (ordered) vector a with N distinct integers where $a[0] < a[1] < \ldots < a[N-1]$. The N integers define $N+1$ non-overlapping ranges $\big(-\infty, a[0]\big]$, $\big(a[0], a[1]\big]$, …, $\big(a[N-1], \infty\big)$ in $\mathbb{R}$. Given a value x, a common task is to determine the range that x falls in. Write down the pseudo code of a $(\log N)$-time algorithm that determines the range. For simplicity, you can assume that x is different from all elements in a. (*Hint: Think about binary search and finding the largest $a[n]$ that is smaller than x*)