Name____________ Student ID_________________ Department/Year________

Final Examination

Introduction to Computer Science
Class#: 901 E10110, Session#: 03
Spring 2014

15:40-17:20 Wednesday
June 18, 2014

Prohibited

1. You are not allowed to write down the answers using pencils. Use only black- or blue-inked pens.
2. You are not allowed to read books or any references not on the question sheets.
3. You are not allowed to use calculators or electronic devices in any form.
4. You are not allowed to use extra sheets of papers.
5. You are not allowed to have any oral, visual, gesture exchange about the exam questions or answers during the exam.

Cautions

1. Check if you get **12** pages (including this title page), **11** questions.
2. Write your name (in Chinese), student ID, and department/year down on top of the cover page.
3. There are in total **100** points to earn. You have **100 minutes** to answer the questions. Skim through all questions and start from the questions you feel more confident with.
4. You are allowed to use **English only** to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with those errors your answers will still make sense.
5. If you have any extra-exam emergency or problem regarding the exam questions, raise your hand quietly. The exam administrator will approach you and deal with the problem.

1. There is a loop in the following algorithm.

```
Latest <- 0;
CurrentValue <- 1;
while (CurrentValue < 50) do
    (print the value assigned to CurrentValue;
     Temp <- Latest;
     Latest <- CurrentValue;
     CurrentValue <- Latest + Temp;)
```

(a) Identify the initialization step(s), the modification step(s), and the test step of the loop. (5%)

(b) Tell the list of numbers the algorithm prints. (5%)

Sample Solution:

(a) Initialization step(s): The first two assignment statements

Modification step(s): The last assignment statement

(Some could argue that it is the last three assignment statements.)

Test step: "while (Current Value< 50)…"

(b) 1, 1, 2, 3, 5, 8, 13, 21, 34.

2. Consider again the following algorithm.

```
Latest <- 0;
CurrentValue <- 1;
while (CurrentValue < 50) do
    (print the value assigned to CurrentValue;
     Temp <- Latest;
     Latest <- CurrentValue;
     CurrentValue <- Latest + Temp;)
```

(a) Rewrite the algorithm using a repeat-until loop than a while-do loop. (5%)

(b) Rewrite the algorithm using a recursive structure than an iterative structure. (10%)

Sample Solution:

(a)

```
Latest <- 0;
CurrentValue <- 1;
repeat
    (print the value assigned to CurrentValue;
     Temp <- Latest;
     Latest <- CurrentValue;
     CurrentValue <- Latest + Temp;)
until (CurrentValue >= 50)
```

(b)

```
procedure MysteryWrite (Lasest, CurrentValue)
if (CurrentValue < 50) then
    (print the value assigned to CurrentValue;
     Temp <- CurrentValue + Lastest;
     MysteryWrite (CurrentValue, Temp))

MysteryWrite (0, 1)
```

3. When searching for an entry within the list:

   L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
   (Note that the list is in alphabetical order.)

   (a) how many entries, including S, will be considered before discovering that the entry is present using the sequential search algorithm and how many for using the binary search algorithm? (5%)
   (b) Which entry to search will result in the worst-case performance for the sequential search algorithm and which entry to search will result in the worst-case performance for the binary search algorithm? (5%)

Sample Solution:

(a) sequential search: 8; binary search: 1

(b) worst case for sequential search: Z (15 entries considered)
    worst case for binary search: L, N, P, R, T, V, X, or Z (4 entries considered)

4. Summarize the following rat's-nest routine with a single if-then-else statement. (5%)

```
         if X > 15 then goto 60
         X = X - 9
         goto 110
     60  X = X * 7
    110  Stop
```

Sample Solution:

```
if X > 15 then
    X = X * 7
else
    X = X - 9
```

5. Suppose the procedure Modify is defined as follows.

```
Procedure Modify (Y) {
  Y <- 9;
  Print the value of X;
  Print the value of Y;}
```

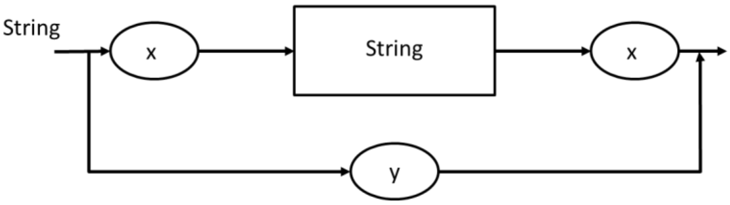
Suppose that X is a global variable and the following program segment is executed.

```
X <- 5;
Modify (X);
Print the value of X;
```

What will be printed if the parameters are passed by value and what if the parameters are passed by reference? (5%)

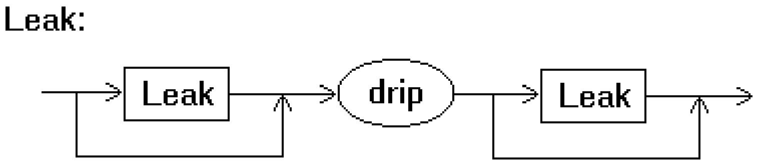
Sample Solution:
Pass by value: 5, 9, 5
Pass by reference: 9, 9, 9

6. Describe the structure of the possible strings coming out of the String diagram below. (5%)



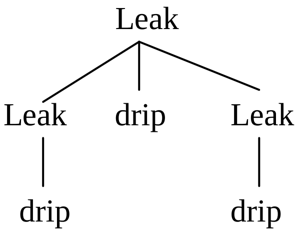
Sample Solution:
$x^n y x^n$, where n is a nonnegative integer

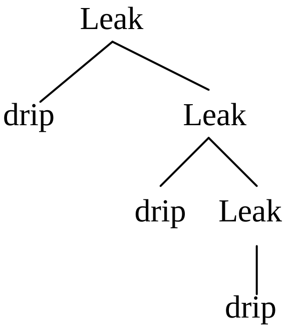
7. Show that the grammar below is ambiguous by drawing two distinct parse trees for the string “drip drip drip.” (10%)



Sample Solution:

Possible answers include:





8. For the well-known searching and sorting problems, we learn 2 algorithms solving for each. The efficiency of insert sort is O(N^2), merge sort O(NlgN), sequential search O(N), and binary search O(lgN). Sort and list the 4 algorithms by the efficiency as N approaches ∞, from the fastest to the slowest. (5%)

Sample Solution:

binary search, sequential search, merge sort, insert sort.

9. Which one(s) of the following statement is(are) correct? (5%)

(a) The searching problem is computable.

(b) The sorting problem is computable.

(c) The halting problem is computable.

(d) The halting problem is a P class problem.

(e) The halting problem is a NP-complete problem.

Sample Solution:

(a) (b)

10. Show how the statement could be simulated in Bare Bones. (10%)

```
name3  <- name1 - name2;
```

Sample Solution:

```
clear aux1;
while (name2 not 0) {
   decr name2;
   decr name1;
   incr aux1;}
clear name3, aux2;
while (name1 not 0) {
   decr name1;
   incr name3;
   incr aux2;}
while (aux1 not 0) {
   decr aux1;
   incr name1;
   incr name2;}
while (aux2 not 0) {
   decr aux2;
   incr name1;}
```

11. Draw the search tree to solve the eight-puzzle from the following start state.

```
 13
425
786
```

(a) Use the heuristic search by searching only the node with the lowest number of moves of all tiles to reach their rightful positions. (10%)

(b) User the breadth-first search without any heuristics. (10%)

Sample Solution:

The tree might vary slightly from student to student. As long as the derivation makes sense, points will be awarded.