| COS 126 | General Computer Science | Fall 2017 |
|---|---|---|

# Written Exam 2

This exam has 7 questions (including question 0) worth a total of 70 points. You have 50 minutes. Write all answers inside the designated spaces.

**Policies.** The exam is closed book, except that you are allowed to use a cheatsheet (8.5-by-11 paper, two sides, in your own handwriting). No electronic devices are permitted.

**Discussing this exam.** Discussing the contents of this exam before solutions have been posted is a violation of the Honor Code.

**This exam.** Do not remove this exam from this room. Write your name, NetID, and the room in which you are taking the exam in the space below. Mark your precept number. Also, write and sign the Honor Code pledge. You may fill in this information now.

**Name:**

**NetID:**

**Exam Room:**

**Precept:**

*"I pledge my honor that I will not violate the Honor Code during this examination."*

*Signature*

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | total |
|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |

0. **Miscellaneous. (1 point)**

   (a) Write your name, NetID, Exam Room, and Precept Number in the space provided on the front of this exam.

   (b) Write and sign the Honor Code pledge on the front of this exam.

1. **Representations. (15 points)**

   (a) Express the decimal number 36 as:

   i. a binary number:

   ii. a hexadecimal number:

   (b) In ASCII (the character representation used by almost all modern computers), the character ‘P’ is represented by the binary number 1010000. Express the ASCII character ‘U’ as a **decimal number**.

   (c) Express the decimal number -27 in 8-bit two’s complement.

   (d) What is the sum of 8F2D and 707A (both hexadecimal) in hexadecimal format?

2. **Boolean Circuits. (9 points)**

Consider a circuit that has three inputs ($x$, $y$, and $z$) and produces an output ($f$) of true (1) if and only if exactly one of the inputs is true (1).

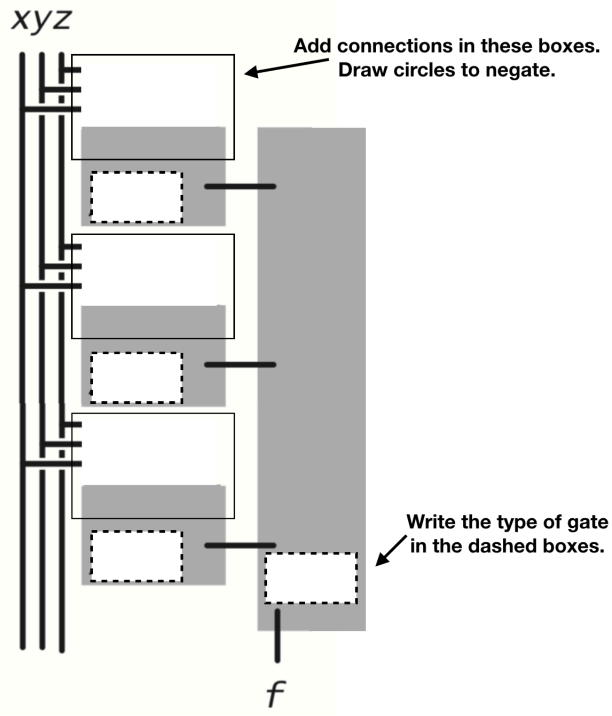
(a) Complete the truth table for this circuit:

| $x$ | $y$ | $z$ | $f$ |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

(b) Write out the sum-of-products formula for this circuit. (Do not simplify.)

$f =$

(c) Draw a circuit for $f$ by filling in the solid and dashed lined boxes.



Add connections in these boxes. Draw circles to negate.

Write the type of gate in the dashed boxes.

3. **Tracing a TOY Program. (12 points)**

Consider the following TOY program.

```
10: 7202
11: 8199
12: 4312
13: 1A02
14: 7C00
15: CA19
16: 1CC3
17: 2AA1
18: C015
19: 9AFF
1A: 9CFF
1B: 0000

99: 0001
```

(a) After executing the instructions at **memory locations 10, 11, and 12**, what are the values of register 1, 2, and 3 in hexadecimal?

| Register: | R[1] | R[2] | R[3] |
|---|---|---|---|
| Value: | | | |

(b) What does this program print?

## TOY REFERENCE CARD

```
INSTRUCTION FORMATS

               | . . . . | . . . . | . . . . | . . . .|
  Format RR:   | opcode  |    d    |    s    |    t    |  (1-6, A-B)
  Format A:    | opcode  |    d    |       addr        |  (7-9, C-F)


ARITHMETIC and LOGICAL operations
    1: add              R[d] <- R[s] +  R[t]
    2: subtract         R[d] <- R[s] -  R[t]
    3: and              R[d] <- R[s] &  R[t]
    4: xor              R[d] <- R[s] ˆ  R[t]
    5: shift left       R[d] <- R[s] << R[t]
    6: shift right      R[d] <- R[s] >> R[t]

TRANSFER between registers and memory
    7: load address     R[d] <- addr
    8: load             R[d] <- M[addr]
    9: store            M[addr] <- R[d]
    A: load indirect    R[d] <- M[R[t]]
    B: store indirect   M[R[t]] <- R[d]

CONTROL
    0: halt             halt
    C: branch zero      if (R[d] == 0) PC <- addr
    D: branch positive  if (R[d] >  0) PC <- addr
    E: jump register    PC <- R[d]
    F: jump and link    R[d] <- PC; PC <- addr


Register 0 always reads 0.
Loads from M[FF] come from stdin.
Stores to  M[FF] go to stdout.

16-bit registers (using two's complement arithmetic)
16-bit memory locations
 8-bit program counter
```

4. **P, NP, and NP-Complete. (12 points)**

   Assume that computer scientists have **proven** that P is not equal to NP. CIRCLE either `TRUE`, `FALSE` or `UNKNOWN` (i.e. statement is still unknown to computer scientists even in light of this proof) given this assumption.

   (a) P is not equal to NP.

   [✓ TRUE] FALSE UNKNOWN

   (b) There does not exist an efficient algorithm for finding the non-trivial integer prime factors of a number (FACTOR).

   TRUE FALSE UNKNOWN

   (c) The traveling salesperson problem (TSP) is not in P.

   TRUE FALSE UNKNOWN

   (d) The "Is this list sorted?" decision problem is not in NP.

   TRUE FALSE UNKNOWN

   (e) There exists an efficient algorithm for finding optimal TSP tours, but no one has been able to find it.

   TRUE FALSE UNKNOWN

   (f) SAT poly-time reduces to SORT.

   TRUE FALSE UNKNOWN

   (g) No algorithm can guarantee to efficiently find an optimal tour for TSP.

   TRUE FALSE UNKNOWN

   (h) One can construct a program that accepts an integer $N$, efficiently produces a single TSP of size $N$ of its choice, and efficiently computes that TSP's optimal tour.

   TRUE FALSE UNKNOWN

   (i) There does not exist an efficient algorithm for the Boolean satisfiability problem. (SAT)

   TRUE FALSE UNKNOWN

5. **Program Analysis (9 points)**

In the following `mystery` functions, the array `inArray` of size $N$ is the input. What is the order of growth in terms of $N$ for each `mystery` function?

```
public static int mystery1(int inArray[]) {
    int N = inArray.length;

    int sum = 0;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < i/2; j++)
            sum += inArray[j];

    for (int i = 0; i < N; i++)
        sum += inArray[i];

    return sum;
}
```

What is the order of growth of mystery1? Circle the best answer:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $N$ | $N^2$ | $N^3$ | $N^4$ | $logN$ | $NlogN$ | $N^2logN$ | $2^N$ | $N!$ |

```
public static int mystery2(int[] inArray) {
    int N = inArray.length;

    int sum = 0;
    for (int i = 0; i < 10; i++)
        for (int j = 0; j < N; j++)
            sum += inArray[j];

    return sum;
}
```

What is the order of growth of mystery2? Circle the best answer:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $N$ | $N^2$ | $N^3$ | $N^4$ | $logN$ | $NlogN$ | $N^2logN$ | $2^N$ | $N!$ |

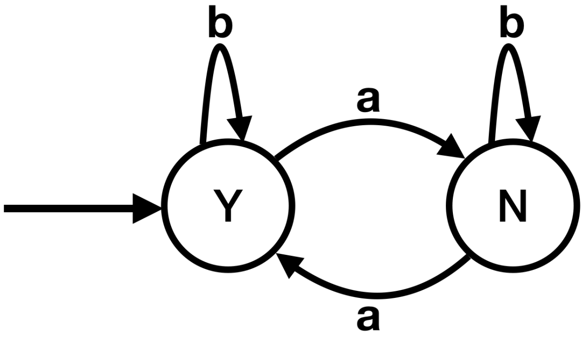
```
/* NOTE: This function is called by mystery3 below. */
public static int mystery3_helper(int[] inArray, int left, int right) {
    int sum = 0;
    int mid = ((right - left) / 2) + left;

    if (left == right) return 0;
    if (left + 1 == right) return 0;

    for (int i = left; i <= right; i++)
        sum += inArray[i];

    return mystery3_helper(inArray, left, mid)  +
        mystery3_helper(inArray, mid, right) +
        sum;
}

public static int mystery3(int[] inArray) {
    int N = inArray.length;
    return mystery3_helper(inArray, 0, N-1);
}
```

What is the order of growth of mystery3 including the time spent in mystery3_helper? Circle the best answer:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | $N$ | $N^2$ | $N^3$ | $N^4$ | $logN$ | $NlogN$ | $N^2logN$ | $2^N$ | $N!$ |

6. **Regular Expressions and Deterministic Finite Automata (12 points)**

Consider the following DFA:



Fill out the table. Write "Yes" in the second column if the description matches the DFA. Write"No" in the second column if the description does not match the DFA. If the description does not match the DFA, prove it by writing a counterexample input string in the third column. The counterexample can be a string that is accepted by the DFA but not by the description, or it can be a string that is accepted by the description but not the DFA. Use $\epsilon$ to denote the empty string.

| Description or RE | Equivalent?<br>(Yes or No) | Counterexample if not<br>equivalent |
|---|---|---|
| All strings over $a, b$ | | |
| All strings over $a, b$ with at least two $a$'s | | |
| `b*a*b*` | | |
| `(b*ab*a)*` | | |
| `b*ab*a` | | |
| `b*(ab*a)*` | | |
| `(b*(ab*a))*` | | |
| `(b*(ab*a)*)*` | | |

This page intentionally left blank.

*This page is provided as scratch paper. If you tear it out, write your name, NetID, and precept number in the space provided and return it inside your exam.*

**Name:** ____________________ **NetID:** __________ **Precept:** ______