## Name:

## Instructions

1. Write your name at the top of the first page and your initials at the bottom of every page.
2. When you are done, return the exam with all the pages arranged in ascending order. Do not staple the exam.
3. This is a closed-book exam. No form of communication is permitted (eg, talking, texting, etc.), except with the course staff.
4. No electronic devices are permitted.
5. There are 25 multiple-choice questions in this exam, each worth 3 points.
6. The answer to each question must be marked with a pencil as shown in the following example. It will be considered incorrect otherwise.

Example. What is Albert Einstein's miracle year?
(A) 1879
(B) 1900
(C) 1905
(D) 1917
(E) 1955
7. You may use the blank spaces for any scratch work.
8. Discussing the exam contents with anyone who has not taken the exam is a violation of the academic honesty code.

Problem 1. Consider the following function:

```
public static int mystery(int[][] a) {
    int x = 0;
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a[0].length; j++) {
            x += (i == j) ? a[i][j] * a[i][j] : 0;
        }
    }
    return x;
}
```

What does mystery (a) return, where $a=\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$ ?
(A) 126
(B) 107
(C) 66
(D) 45
(E) 83

Problem 2. Consider the following recursive function:

```
public static int mystery(int a, int b) {
    return (b == 0) ? a : mystery (b, a % b);
}
```

a. What does mystery $(8,15)$ return?
(A) 1
(B) 7
(C) 120
(D) 15
(E) 8
b. What does mystery $(27,72)$ return?
(A) 216
(B) 9
(C) 27
(D) 72
(E) 45
c. What does mystery() compute and return in general?
(A) Greatest common divisor of $a$ and $b$
(B) $|a-b|$
(C) $a$
(D) $b$
(E) Least common multiple of $a$ and $b$

Problem 3. Consider the following recursive functions:

```
public static int f(Node x) {
    return (x == null) ? 0: 1 + f(x.next);
}
public static int g(Node x) {
    return (x == null) ? 0 : g.item + g(x.next);
}
```

a. What does $\mathrm{f}(\mathrm{a})$ return, where a is a reference to the first node in the linked list containing the items $1,1,2,3,5,8$, and 13 and in that order?
(A) 1
(B) 13
(C) 33
(D) 0
(E) 7
b. What does $\mathrm{g}(\mathrm{a})$ return, where a is a reference to the first node in the linked list containing the items $1,1,2,3,5,8$, and 13 and in that order?
(A) 33
(B) 0
(C) 7
(D) 1
(E) 13

Problem 4. Consider the following program Mystery.java:

```
import stdlib.StdIn;
import stdlib.StdOut;
public class Mystery {
    public static void main(String[] args) {
        String x = StdIn.readString();
        String y = StdIn.readString();
        StdOut.print(x + y);
        StdOut.print(" ");
        StdOut.print (y + x);
        StdOut.println();
    }
}
```

Next, suppose that the file input.txt contains the two strings ab and ba separated by a space. What does the following command output?
\$ java Mystery < input.txt I java Mystery I java Mystery
(A) abbabaabbaababbabaababbaabbabaab baababbaabbabaababbabaabbaababba
(B) abbabaab baababba
(C) ab ba
(D) abbabaabbaababba baababbaabbabaab
(E) abba baab

Problem 5. Consider the following table, which gives the running time $T(n)$ in seconds for a program for various values of the input size $n$ :

| $n$ | $T(n)$ |
| :---: | :---: |
| 1000 | 3 |
| 2000 | 12 |
| 4000 | 48 |
| 8000 | 192 |

a. What is the value of $T(n)$ if $n=16000$ ?
(A) 576
(B) 1536
(C) 192
(D) 384
(E) 768
b. What is the running time classification for the program?
(A) Quadratic
(B) Logarithmic
(C) Linear
(D) Linearithmic
(E) Cubic

Problem 6. What is the running time classification for following code fragment?

```
int sum = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < 100; j++) {
        for (int k=0; k < 1000; k++) {
        sum++;
        }
    }
}
```

(A) Linearithmic
(B) Cubic
(C) Quadratic
(D) Exponential
(E) Linear

Problem 7. Consider a data type t with two instance variables: int x and double y. Ignoring array and object overheads, what is the memory footprint (in bytes) of the array a[] created and initialized as follows?

```
T[] a = new T[100];
for (int i = 0; i < 100; i++) {
    T[i] = new T();
}
```

(A) 1200
(B) 800
(C) 12
(D) 400
(E) 100

Problem 8. Consider the following functions:

```
public static Iterator<Character> f(String s) {
    Queue<Character> Q = new Queue<Character>();
    for (int i = 0; i < s.length(); i++) { Q.enqueue(s.charAt(i)); }
    return Q.iterator();
}
public static Iterator<Character> g(String s) {
    Stack<Character> S = new Stack<Character>();
    for (int i = 0; i < s.length(); i++) { S.push(s.charAt(i)); }
    return S.iterator();
}
```

a. What is the value returned by $f($ "alice") .next()?
(A) $i_{i}$,
(B) ${ }^{\prime}{ }^{\prime}$
(C) ${ }_{1}$,
(D) ${ }^{c}$,
(E) ${ }^{\prime} e^{\prime}$
b. What is the value returned by the method call g ("alice").next()?
(A) ${ }_{1}$,
(B) ${ }_{i}$,
(C) ${ }^{e}{ }^{\prime}$
(D),$c$,
(E) ${ }^{\prime}{ }^{\prime}$

Problem 9. Suppose we use the quickunionuf data structure to solve the dynamic connectivity problem with 10 sites and input pairs $(8,1),(7,6),(9,2),(7,8),(4,6),(6,0)$, and $(4,1)$, arriving in that order; the code for the union() method in QuickUnionuF is shown below.

```
public void union(int p, int q) {
    int rootP = find(p);
    int rootQ = find(q);
    if (rootP == rootQ) {
        return;
    }
    parent[rootP] = rootQ;
    count--;
}
```

a. What are the values in the parent array after all the pairs are processed?
(A) parent $=\{2,0,2,0,0,0,6,0,8,0\}$
(B) parent $=\{0,1,0,2,0,0,0,2,8,0\}$
(C) parent $=\{0,0,2,0,0,2,6,0,8,0\}$
(D) parent $=\{0,0,0,3,2,2,0,0,0,9\}$
(E) parent $=\{0,0,2,3,1,5,1,6,1,2\}$
b. What is the size of the largest component?
(A) 7
(B) 4
(C) 3
(D) 6
(E) 5
c. What is the identifier of the largest component?
(A) 1
(B) 4
(C) 2
(D) 3
(E) 0

Problem 10. Consider sorting an array a[] containing the following strings, using selection sort (shown below):

| C | G | H | Y | V | T | S | M | Z | N |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

```
public static void sort(Comparable[] a) {
    int n = a.length;
    for (int i = 0; i < n; i++) {
        int min = i;
        for (int j = i + 1; j < n; j++) {
            if (less(a[j], a[min])) {
                min = j;
            }
        }
        exchange(a, i, min);
    }
}
```

a. What is the value that $н$ is exchanged with?
(A) N
(B) H
(C) $m$
(D) s
(E) $z$
b. What is the value that y is exchanged with?
(A) m
(B) ${ }^{\mathrm{N}}$
(C) H
(D) s
(E) z

Problem 11. Consider sorting an array a[] containing the following strings, using insertion sort (shown below):

| $D$ | $J$ | $M$ | $R$ | $S$ | $T$ | $Y$ | $Z$ | $D$ | $F$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

```
public static void sort(Comparable[] a) {
    int n = a.length;
    for (int i = 1; i < n; i++) {
        for (int j = i; j > 0 && less(a[j], a[j - 1]); j--) {
            exchange(a, j, j - 1);
        }
    }
}
```

Where is the item o sorted (ie, what is its index) relative to the items before it?
(A) 2
(B) 1
(C) 0
(D) 3
(E) 4

Problem 12. Consider sorting an array a[] containing the following strings (already shuffled), using quick sort (shown below):

| V | U | Z | L | S | Y | R | E | I | J |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

```
public static void sort(Comparable[] a) {
        StdRandom.shuffle(a);
        sort(a, 0, a.length - 1);
}
private static void sort(Comparable[] a, int lo, int hi) {
    if (hi <= lo) {
        return;
    }
    int j = partition(a, lo, hi);
    sort(a, lo, j - 1);
    sort(a, j + 1, hi);
}
private static int partition(Comparable[] a, int lo, int hi) {
    int i = lo;
    int j = hi + 1;
    Comparable v = a[lo];
    while (true) {
        while (less(a[++i], v)) {
            if (i == hi) {
                break;
            }
        }
        while (less(v, a[--j])) {
            if (j == lo) {
                break;
            }
        }
        if (i >= j) {
                break;
        }
        exchange(a, i, j);
    }
    exchange(a, lo, j);
    return j;
}
```

a. What is the destination index of the pivot element after the first call to partition()?
(A) 7
(B) 6
(C) 8
(D) 5
(E) 9
b. What is the state of the array a after the first call to partition()?
(A) E J J I
(B) I $\quad$ E $\quad \mathrm{J} \quad \mathrm{R} \quad \mathrm{U} \quad \mathrm{S} \quad \mathrm{L} \quad \mathrm{V} \quad \mathrm{Z} \quad \mathrm{Y}$
(C) I $\quad$ I $\quad$ U $\quad$ R $\quad$ J $\quad$ L $\quad$ E $\quad$ V $\quad$ Y $\quad$ Z
(D) $\begin{array}{lllllllllll} & R & L & U & J & E & I & S & V & Z & Y\end{array}$

c. What is pivot element in the next call to partition()?
(A) $R$
(B) $Y$
(C) E
(D) z
(E) I

Problem 13. Insert the following keys in that order into a max-heap:
$\begin{array}{lllllllll}B & Z & Q & K & V & F & S & N & I\end{array}$
a. What is the index of the key ?
(A) 5
(B) 4
(C) 7
(D) 6
(E) 8
b. What is the key with index 6 ?
(A) $F$
(B) $Q$
(C) s
(D) $\kappa$
(E) ${ }^{\mathrm{N}}$
c. If we perform a delmax() operation on the tree, what is the key that will replace the current maximum before it is sunk down?
(A) $F$
(B) I
(C) $Q$
(D) $\kappa$
(E) в

## Answers

Problem 1. B

Problem 2. A, B, A
Problem 3. E, A
Problem 4. D
Problem 5. E, A
Problem 6. E

Problem 7. A

Problem 8. B, C
Problem 9. E, D, E
Problem 10. B, A
Problem 11. D
Problem 12. A, E, C
Problem 13. E, A, B

