

Name _____

Computer Science E-22 Practice Midterm

This exam consists of two parts. Part I has 5 multiple-choice questions worth 3 points each. Part II consists of 3 problems; show all your work on these problems so that partial credit may be awarded if your final answer is wrong but your reasoning is partially correct.

You have one hour to complete the exam. The questions are worth a total of 50 points. In order to properly budget your time, plan on spending approximately one minute per point.

You may use any notes, books, or other references at your disposal. However, do not waste too much time leafing through your materials. You may *not* use a personal computer or other computing device. **Do all your work on the exam itself.**

Write clearly. Good luck!

Problem	Max. Score	SCORE
I	15	
II-1	12	
II-2	11	
II-3	12	
TOTAL	50	

Answers to Part I:

question #:	1	2	3	4	5
first choice:					
second choice:					

Part I. Multiple-choice (3 pts. each). Specify up to two answers.

Write up to two answers in the table at the bottom of the cover sheet. You will get one point for any problem where your first choice is wrong but your second choice is correct.

1. The following array is to be sorted in ascending order:

25 56 18 20 12 9 15

Which algorithm will cause the array to be ordered

18 20 12 9 15 25 56

at an intermediate stage in the sorting process?

- A. Shell sort (initial increment = 3)
- B. insertion sort
- C. bubble sort
- D. quicksort (initial pivot = 20)
- E. selection sort

2. The following method/function sorts an array of floating-point numbers:

```
public static void sort(double a[]) {
    int i, j, n;
    double temp;

    n = a.length;
    for (i = 1; i < n; i++) {
        temp = a[i];
        j = i;
        while (j >= 1 && a[j-1] > temp) {
            a[j] = a[j-1];
            j--;
        }
        a[j] = temp;
    }
}
```

In the worst case, the number of times that the condition `a[j-1] > temp` is evaluated grows in proportion to what function of n ?

- A. n
- B. $n!$
- C. $n \cdot \log(n)$
- D. $\log(n)$
- E. n^2

3. Consider the following function/method:

```
int test(int a, int b) {
    if (a < b)
        return 0;
    else
        return (1 + test(a-b, b));
}
```

What is returned by the call `test(15, 4)`?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

4. Given the array of integers `arr` shown below

13	7	27	2	18	33	9	11	22	8
----	---	----	---	----	----	---	----	----	---

what is the output of the following statements?

```
int[] p = new int[10];
int[] q = new int[10];
for (int i = 0; i < 10; i++)
    p[i] = arr[i];
q = p;
p[4] = 20;
System.out.println(arr[4] + " " + q[4]);
```

- A. 20 20
- B. 18 18
- C. 2 2
- D. 18 20
- E. 2 20

5. You have a singly linked list constructed out of nodes defined as follows:

```
public class Node {
    public int datum;
    public Node next;
}
```

In all of functions shown below, the parameter `first` refers to the first node in the linked list, if there is one, and has the value `null` otherwise. Which of the following functions correctly inserts a value `x` at the front of the linked list and returns a reference to the new front of the linked list?

- I.

```
public Node insertFront(Node first, int x) {
    first = new Node();
    first.datum = x;
    first.next = first;
    return first;
}
```
- II.

```
public Node insertFront(Node first, int x) {
    Node n = new Node();
    n.datum = x;
    n.next = first;
    return n;
}
```
- III.

```
public Node insertFront(Node first, int x) {
    Node n = new Node();
    first = n;
    n.datum = x;
    n.next = first;
    return first;
}
```

- A. I only
- B. II only
- C. III only
- D. I and II
- E. II and III

PART II: Answer all three questions in the space provided.

II-1. Sorting (12 points total; 3 points per part)

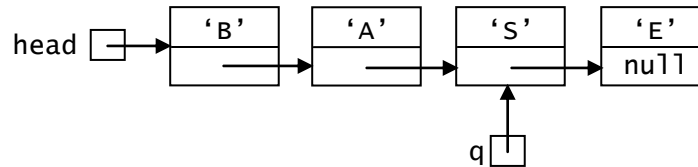
The array below is to be sorted in ascending order.

17 53 71 62 36 46 41 23 12

- After the initial partition step of the version of quicksort discussed in lecture, with 36 as the pivot, how would the array be ordered?
- Assume that the initial iteration of Shell sort uses an increment of 3. After that initial iteration, how would the array be ordered?
- After the initial iteration of bubble sort, how would the array be ordered?
- On how many of the passes of insertion sort would the inner loop be skipped? Explain your answer briefly.

II-2. Linked Lists (11 points total)

The diagram below shows a linked list of characters, along with two variables that each store a reference/pointer to one of the nodes in the list.



The nodes in the linked list could be implemented using the following class:

```

public class CharNode {
    public char val;
    public CharNode next;
}

```

a. 2 points

On the diagram, circle the memory location specified by the expression `head.next.next`

b. 2 points

What is the *value* of the expression from part a?

c. 3 points

Write one or more lines of code that remove the node containing the character 'S' from the list.

d. 4 points

Modify the diagram above to reflect the results of executing the following lines on the original version of the list (before the 'S' was removed):

```

q = q.next;
q.next = head;

```

II-3. Recursion and Algorithm Analysis (12 points total)

a. 8 points

Write a *recursive* method named `sumReciprocals` that takes as its only argument a non-negative integer, `n`, and returns a double value that is the sum of the reciprocals of the integers from 1 to `n`. For example, `sumReciprocals(2)` should return 1.5, which is $1/1 + 1/2$, and `sumReciprocals(4)` should return approximately 2.0833, which is $1/1 + 1/2 + 1/3 + 1/4$. You do *not* need to perform any error-checking on the value of the parameter. No use of iteration is allowed.

```
public static double sumReciprocals(int n) {
```

```
}
```

b. 4 points

Give a big-O expression for the number of method calls that are made when using your answer from part (a) to evaluate `sumReciprocals(n)`.

Supplemental Practice Problems

S-I. Multiple Choice

1. Which of the following statements is *not* true about the increments used in Shell sort?

- A. The increments should be decreasing.
- B. The increments should be relatively prime.
- C. The final increment should be 1.
- D. The increments should be prime numbers.
- E. All the statements above are true.

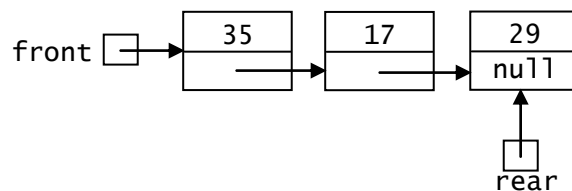
2. Here is an array that has just been partitioned by the first step of quicksort:

3 0 2 4 5 8 7 6 9

Which of the following statements is correct?

- A. 5 could be the pivot, but 7 could not be.
- B. 7 could be the pivot, but 5 could not be.
- C. Neither 5 nor 7 could be the pivot.
- D. Either 5 or 7 could be the pivot.

3. The diagram below suggests how we could implement a double-ended linked list, in which we maintain a reference to both the first and last nodes in the linked list.



Which one of the following operations would be inefficient to carry out when there are a large number of elements in the linked list?

- A. insertion at the end to which front refers
- B. insertion at the end to which rear refers
- C. deletion from the end to which front refers
- D. deletion from the end to which rear refers
- E. test for an empty linked list

4. You have a singly linked list constructed out of nodes defined as follows:

```
public class Node {  
    public int datum;  
    public Node next;  
}
```

In the function shown below, the parameter `f` refers to the first node in the linked list, if there is one, and has the value `null` otherwise. The intent of the function is to remove the last node of the linked list.

```
public void removeLast(Node first) {  
    Node p, q;  
  
    p = first;  
    q = p.next;  
  
    while (q.next != null) {  
        p = q;  
        q = q.next;  
    }  
  
    p.next = null;  
}
```

Which of the following describes the class of linked lists for which this function works correctly?

- A. No linked lists
 - B. All nonempty linked lists
 - C. All linked lists with more than one node
 - D. Empty linked lists and all linked lists with more than one node
 - E. All linked lists
5. Through experiment, you determine that **selection sort** performs 5000 **comparisons** when sorting an array of some size k . If you doubled the size of the array to $2k$, approximately how many comparisons would you expect it to perform?
- A. 5000
 - B. 10000
 - C. 20000
 - D. 40000
 - E. the value would depend on the contents of the array

6. Through experiment, you determine that **selection sort** performs 5000 **moves** when sorting an array of some size k . If you doubled the size of the array to $2k$, approximately how many moves would you expect it to perform?

- A. 5000
- B. 10000
- C. 20000
- D. 40000
- E. the value would depend on the contents of the array

7. A program for recursive backtracking includes a method similar to this one:

```
void key_function(int i) {  
    for (int alternative = i; alternative < n;  
        alternative++)  
    {  
        do_alt(alternative);  
        *****  
        undo_alt(alternative);  
    }  
}
```

The line `*****` should be replaced by:

- A. `do_alt(alternative + 1);`
- B. `key_function(alternative);`
- C. `key_function(alternative + 1);`
- D. `key_function(i);`
- E. `key_function(i + 1);`

S-II-1. Sorting

- [illegible]

- b. Let's say that you need to sort a collection of data that is stored in a singly linked list. Would Shellsort be a good algorithm to adapt for this problem? Why or why not?

S-II-2. Recursion

The Fibonacci sequence is a famous sequence of integers that begins as follows:

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

The terms in this sequence can be defined as follows:

- (i) term 0 is 0
- (ii) term 1 is 1
- (ii) all subsequent terms in the sequence can be obtained by summing the previous two terms in the sequence:

$$\text{term}_n = \text{term}_{n-1} + \text{term}_{n-2} \quad \text{for all } n \geq 2$$

For example:

$$\text{term}_3 = \text{term}_2 + \text{term}_1 = 1 + 1 = 2$$

$$\text{term}_5 = \text{term}_4 + \text{term}_3 = 3 + 2 = 5$$

Write a recursive method for finding the nth term in the Fibonacci sequence:

```
public static int fib(int n) {
```

```
}
```