CS 202: Introduction to Computation Fall 2011: Exam #2

Name: Answer Key

Question	Possible Points	Received Points
1	20	
1	20	. Anger
2	20	
2		
3	20	
4	20	
5	20	
	1	-
Total	100	£ = -

This exam is closed notes.

You have 50 minutes to complete the 5 questions on this exam.

Please write your answers clearly.

Good luck!

1 pt each

Question 1: Truth or Consequences

Consider whether each of the following statements is True or False. Circle the correct answer.

True False With an unsorted list, a linear search must look through every element of the list to find the maximum element

True False A binary search assumes that the keys in the list are sorted

True (False) A binary search assumes that the key being searched for exists in the list

True False In a game of twenty questions, the best guessing strategy allows one to identify on the order of 20^2 different items

True False A recursive algorithm can be defined by a base case and a set of rules reducing other cases towards the base case

True False An insertion sort requires O(N²) operations

True False | Merging two lists of size N into a list of size 2*N requires O(N log N) operations

True False A successful merge of the list "2 6 8 9" with the list "4 7 10 11" is "2 4 6 7 8 9 10 11"

True (False) For a very large number of keys, Selection Sort is usually faster than Merge Sort.

True (False) For Quicksort, the best pivot is the key with the maximum value in the list.

True False In Quicksort, a correct partition of the List "8 9 3 2 0 7 2 1" around the pivot 7 is "3 2 0 2 1 7 8 9"

True False) For very large values of N, an algorithm that requires O(N²) operations is probably faster than one that requires O(N log N) operations

True False Web browsers contact web servers with a special type of address called a URL

True (False) A correct simulation computes the same output regardless of the initial conditions

True(False) With cryptography, clear text is confidentially sent over an unprotected network

True False) The complexity of solving a problem is equivalent to the complexity of creating that problem

True) False The algorithm for finding a minimal spanning tree is greedy

True False The minimal spanning tree of a weighted graph is the path a traveling salesperson should use to minimize his or her distance

True False An algorithm that requires 2N⁵ steps has complexity represented by O(N⁵)

True False) All computational problems can be solved in polynomial time

"It is known that

either

accepted

Question 2: List your complaints here

2 pts

Imagine you have eight different Scratch programs, each which manipulates a list "Mystery List" Your job is to "execute" the scripts in your head to repeat the exact same steps and operations that Scratch would. For each of the following 8 scripts, show the contents of "Mystery List" at the end of the script. Be careful of the tiny but important differences across scripts!

```
when A clicked
delete all of Mystery List
 set | Counter | to | 0
 repeat 3
 add Counter to Mystery List -
  set Counter to 5
 when 🦱 clicked
delete all ▼ of Mystery List ▼
 set Counter to 0
 repeat 3
 add Counter to Mystery List
  change Counter by 5
when 🔎 clicked
delete all ▼ of Mystery List ▼
set Counter to 0
repeat length of Mystery List
 add Counter to Mystery List
  change Counter by 2
when A clicked
delete (all ♥) of Mystery List ▼
set Counter to 0
 epeat 3
 add Counter to Mystery List
 add Counter + 1 to Mystery List v
 change Counter by 5
when A clicked
delete ali v of Mystery List v
set Counter to 0
repeat 3
                                                                             10
add Counter to Mystery List
 change Counter by 5
et Counter to 0
 add Counter + 1 to Mystery List
 change Counter by 5
```

```
when A clicked
delete all ▼ of Mystery List ▼
set Counter ▼ to 0
                                                6 4 2 0
repeat 4
 insert Counter at 1 of Mystery List
 change Counter ▼ by 2
when 🦱 clicked
delete all ▼ of Mystery List ▼
set Counter v to 0
repeat 4
                                                   0246
 insert Counter at last ▼ of Mystery List ▼
 change Counter by 2
repeat 2
 delete last ▼ of Mystery List ▼
when A clicked
                                            $ $ 10 15
thing thing
delete all ▼ of Mystery List ▼
set Counter v to 0
repeat 4
add Counter to Mystery List
 change Counter by 5
set Counter▼ to 1
repeat 2
replace item Counter of Mystery List with thing
 change Counter v by 1
```

Question 3: Searching doesn't have to be difficult

Imagine that the Green Flag is clicked and the script "Create Valuable Numbers" creates the two Lists "Valuable Numbers" and "Names" shown below.

```
when clicked
broadcast Create Valuable Numbers and wait
ask What number should I search for and wait
set Key to answer
broadcast Find Key and wait
if not Key Index = 0

say join join The key Key join is located at Index Key Index for a secs
say join join The key Key was not found!

when I receive Find Key
set index to 1
set Key Index to 0

repeat length of Valuable Numbers if item index of Valuable Numbers if item index of Valuable Numbers item key Index item index of Valuable Numbers item index item
```

Va	luable Numbe	rs		Names	
1	884		1	uwygr	
2	608		2	cjeut	
3(734		3	ijuhg	
4	558		4	vxxao	
5	785		5	aldnj	
6	411		6	qoemk	
7	532		7	woxku	
8	331		8	uwiut	
9	545		9	vbpjq	
10	469		10	agtxl	
11	744		11	rkqtp	
12	446		12	vgarn	
13	132		13	vnemo	
14	67		14	bpbmg	
15	166		15	ckhgs	
+	length: 15	1	+	length: 15	

If Key is set to each of the following numbers, what will the scripts "say" to the user? (In other words, what will be the output of the program?) In each case, how many items in Valuable Numbers list will be accessed (or examined)? In each case, how many items in the Names list will be accessed (or examined)?

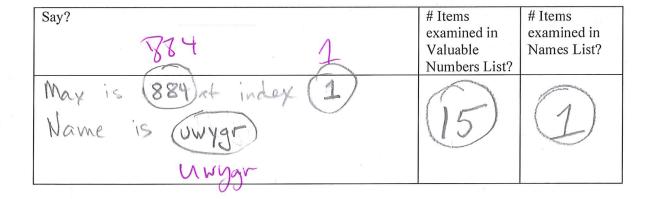
Key Input	Say?	# Items	# Items
		examined in	examined in
		Valuable	Names List?
		Numbers	
	E	List?	1
785	Key: (7.85) at index		
	Matching rame: Gldn i)	5	1
166	Key: 160 at indexi 15	400000000000000000000000000000000000000	-4
	Matchine name: ckhas	15	
121	Key I'll not found		
		15	
734	Key 734 at index 3		1
	Name: 1- wha	3	

4333

Imagine that a similar script "Find Max" is run instead with the same Valuable Numbers and Names lists.

What will the scripts "say" to the user? How many items in Valuable Numbers list will be accessed? How many items in the Names list will be accessed?

```
when index of Valuable Numbers viset Max index vio item Index of Valuable Numbers viset Max index vio item Index of Valuable Numbers viset Max index vio item Index of Valuable Numbers viset Max index vio item Index of Valuable Numbers viset Max index vio index item Index of Valuable Numbers viset Max index vio index item Index of Valuable Numbers viset Max index vio index item Index of Valuable Numbers viset Max index vio index vio index violation index viset Max index vio index violation index viset Max index vio index violation index viset Max index
```



What assumptions does this implementation of "Find Max" make about the list of Valuable Numbers?

2 Rts Contains at least one element > 0

5 pts

Question 4:

Consider the following implementations of Selection Sort:

```
when I receive | Sort List v
 set | ▼ to 1
 repeat until (I)>(List Length)
   set Min v to item I of Unsorted List v
   set Index of min ▼ to 1
   set J▼ to 1
   repeat until (1) > (List Length)
          (item | of Unsorted List ▼ ) < min
       set Min v to item 1 of Unsorted List v
      set index of min v to j
     change Jv by 1
   replace item Index of min of Unsorted List with item I of Unsorted List v
   replace item | of | Unsorted List | with | min
   change | ▼ by 1
and Insertion Sort:
```

```
when I receive Sort List▼
set | ▼ to 1
repeat until (1)> (length of Unsorted List 🔻
 set Item v to item I of Unsorted List v
 delete | of Unsorted List ▼
 set Jy to I
 repeat until (item 1)- 1 of Unsorted List > < Item or 1 = 1
  change Jv by -1
 insert Item at 1 of Unsorted List v
 change | w by 1
```

Assume Unsorted List begins by containing the following 10 integers in this order:

U	Insorted Lis
1	31
2	89
3	21
4	93
5	44
6	87
7	59
8	19
9	43
10	91

Now, either Selection Sort or Insertion Sort is started on the list of 10 integers. The problem is, we don't know which algorithm is being used! The sorting algorithm is then stopped after **each** iteration of the "repeat until" outer loop (that is, just after the last block "change i by 1" increments i to some value) and the contents of the Unsorted List are displayed. We've taken 10 screenshots of Unsorted List over time lettered (a) - (j).

The following page shows the contents of each of the Unsorted Lists, but they have been all mixed up! That is, the picture (a) of Unsorted List could have happened after picture (b) of Unsorted List. Your job is to figure out the correct order!

1. Using the contents of the Unsorted Lists and your knowledge of the algorithms, was Selection or Insertion sort used?

Insertion Sort

2. Order the pictures of Unsorted List to show the sequence in which they must have occurred when the algorithm was run. Write the correct order here.



3. You will notice that some of the Unsorted Lists from different iterations are identical; explain why this is the case!

The keys are not moved because

the next key in the unsarted

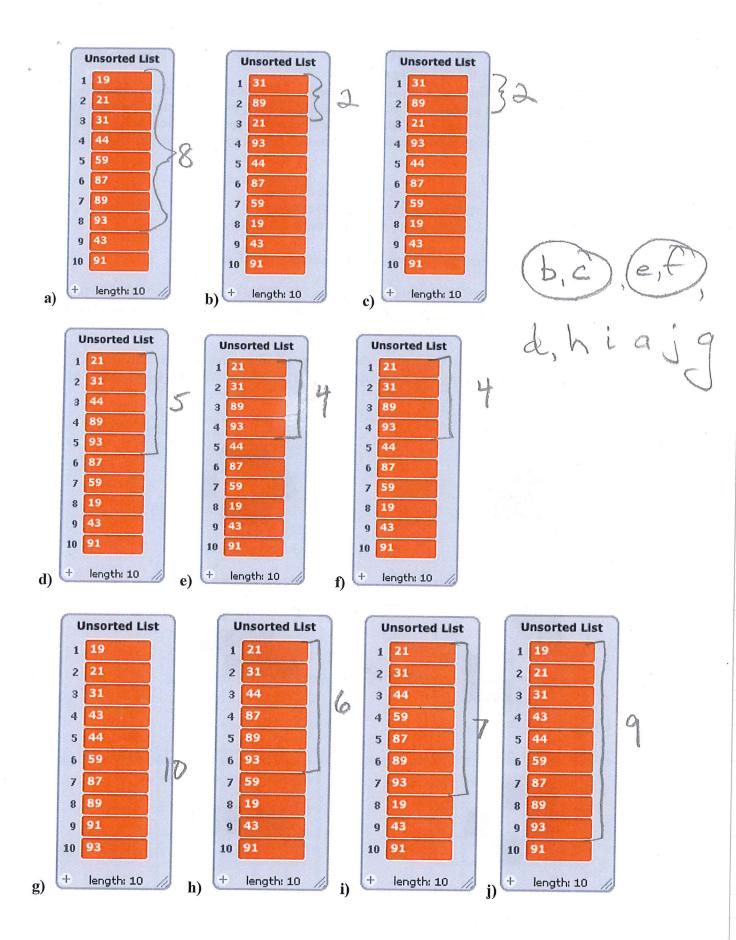
partion of the list is in its correct

position for the next iteration it is the

greatest key in the sorted partion of list.

7 pts

10 pts



-4 pts for any world in list that should not be or omitting world

Question 5: What is the meaning of Life?

The Game of Life is a simulation in which cells are in a 2-D grid; each cell can be either alive (black) or dead (white). The next generation of cells is calculated from the previous generation using a set of rules. Each cell will be alive or dead in the next generation depending upon the current state of its 8 nearest neighbors (the 8 nearest neighbors are the cells directly adjacent above, below, left, right, and the four diagonal cells).

If (cell is alive)

If < 2 neighbors are alive, then the cell dies

If > 3 neighbors are alive, then the cell dies

If 2 or 3 neighbors are alive, then the cell stays alive

If (cell is dead)

if 3 neighbors are alive, then the cell becomes alive

Consider the following six (6) different worlds lettered A-F that each has the current state shown. Some of the shown worlds are stable and will remain constant across future generation while others of the shown worlds will change in some way across generations (e.g., die out, move, or oscillate).

Which worlds are stable and will remain identical through all generations? To help you with your reasoning, you may want to label each cell with its number of live neighbors. Write the letters of the stable worlds here:

