

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

Name: \_\_\_\_\_

Question	Possible Points	Received Points
1	20	
2	20	
3	20	
4	20	
5	20	
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!**

### 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  $O(N^2)$  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  $O(N^2)$  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  $2N^5$  steps has complexity represented by  $O(N^5)$
- True False All computational problems can be solved in polynomial time

## Question 2: List your complaints here

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 green flag clicked
delete all of Mystery List
set Counter to 0
repeat 3
  add Counter to Mystery List
  set Counter to 5
```

```
when green flag clicked
delete all of Mystery List
set Counter to 0
repeat 3
  add Counter to Mystery List
  change Counter by 5
```

```
when green flag 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 green flag clicked
delete all of Mystery List
set Counter to 0
repeat 3
  add Counter to Mystery List
  add Counter + 1 to Mystery List
  change Counter by 5
```

```
when green flag clicked
delete all of Mystery List
set Counter to 0
repeat 3
  add Counter to Mystery List
  change Counter by 5
set Counter to 0
repeat 3
  add Counter + 1 to Mystery List
  change Counter by 5
```

```
when clicked
  delete all of Mystery List
  set Counter to 0
  repeat 4
    insert Counter at 1 of Mystery List
    change Counter by 2
```

```
when clicked
  delete all of Mystery List
  set Counter to 0
  repeat 4
    insert Counter at last of Mystery List
    change Counter by 2
  repeat 2
    delete last of Mystery List
```

```
when clicked
  delete all of Mystery List
  set Counter 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 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.

The image shows a Scratch script and two lists. The script starts with a 'when green flag clicked' event, followed by a 'broadcast Create Valuable Numbers and wait' block, an 'ask What number should I search for? and wait' block, and a 'set Key to answer' block. Then it broadcasts 'Find Key and wait'. An 'if not Key Index = 0' block contains two 'say' blocks: one saying 'The key Key is located at index Key Index for 2 secs' and another saying 'The matching name is item Key Index of Names'. An 'else' block contains a 'say' block saying 'The key Key was not found!'. Below this is a 'when I receive Find Key' block, which sets 'Index to 1' and 'Key Index to 0', then repeats 'length of Valuable Numbers' times. Inside the repeat loop, an 'if item Index of Valuable Numbers = Key' block sets 'Key Index to Index' and 'stop script'. Finally, it changes 'Index by 1'.

The 'Valuable Numbers' list contains 15 items: 884, 608, 734, 558, 785, 411, 532, 331, 545, 469, 744, 446, 132, 67, 166.

The 'Names' list contains 15 items: uwygr, cjeut, ljuhg, vxxao, aldnj, goemk, woxku, uwlut, vbpjg, agtxl, rkqtp, vgarn, vnemo, bpbmg, ckhgs.

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 examined in Valuable Numbers List?	# Items examined in Names List?
785			
166			
121			
734			

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 clicked
broadcast Find Max and wait
say join join The Max element is Max join at index Max Index for 2 secs
say join The winning name is item Max Index of Names
  
```

```

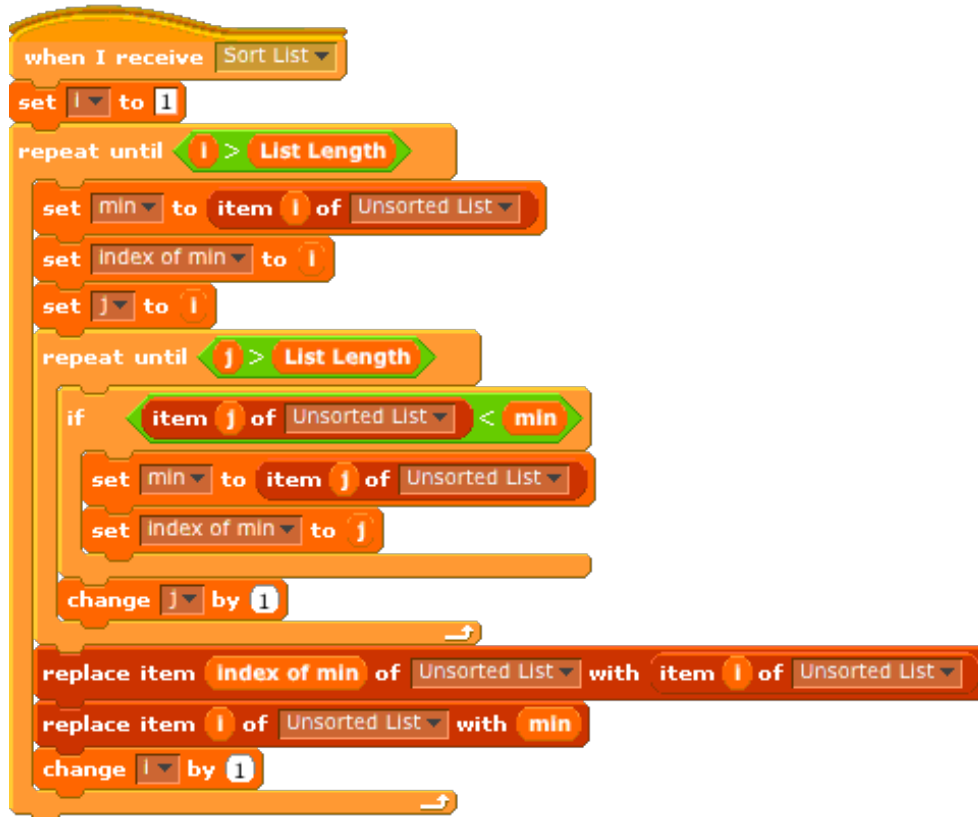
when I receive Find Max
set Max to 0
set Index to 1
set Max Index to 0
repeat length of Valuable Numbers
  if item Index of Valuable Numbers > Max
    set Max to item Index of Valuable Numbers
    set Max Index to Index
  change Index by 1
  
```

Say?	# Items examined in Valuable Numbers List?	# Items examined in Names List?

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

#### Question 4:

Consider the following implementations of Selection Sort:



and Insertion Sort:



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



The image shows a window titled "Unsorted List" with a list of 10 integers. The integers are: 31, 89, 21, 93, 44, 87, 59, 19, 43, and 91. The list is displayed in a vertical column with indices 1 through 10 on the left. At the bottom of the window, it says "length: 10".

Index	Value
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?
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!





a)



b)



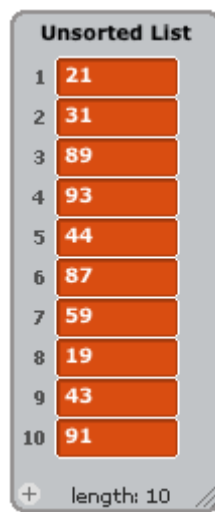
c)



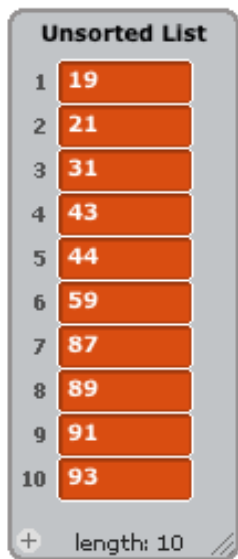
d)



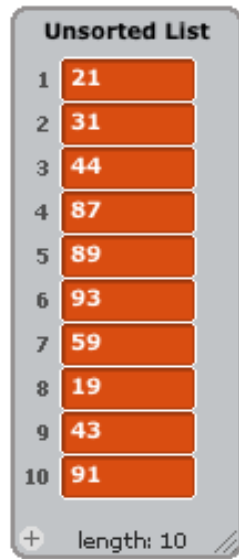
e)



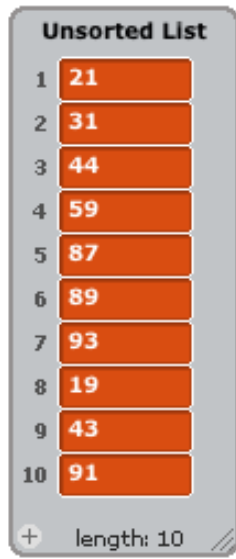
f)



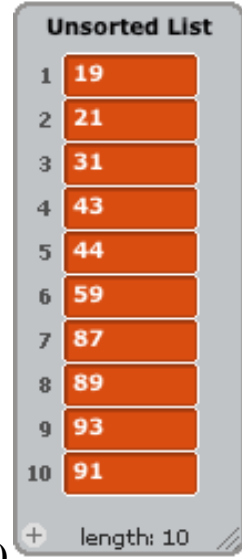
g)



h)



i)



j)

### 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:

