Unit 4 – ICS4U – Arrays & Algorithms

Sample Test, Wednesday May 8, 2023

Name: Gorski

Total	Knowledge	Communication	Thinking	Application
(96)	(22)	(24)	(26)	(24)

Knowledge

1. For each sorting algorithm, colour in the 2 apples that trade places in the first swap. /3

(a) Bubble sort: 4 7 3 1 8 2 5

(b) Selection sort: 4 7 3 1 8 2 5

(c) Quicksort: 4 7 3 1 8 2 5

2. Sort each array using the algorithm indicated. /8

Bubble sort:

6	5	4	7	2
5	6	4	7	2
5	4	6	7	2
5	4	6	2	7
4	5	6	2	7
4	5	2	6	7
4	2	5	6	7
2	4	5	6	7

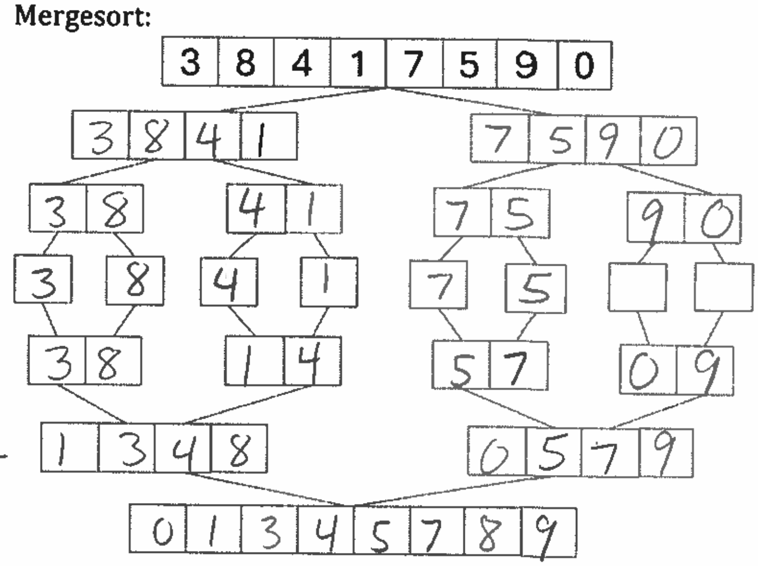
Selection Sort:

6	1	9	5	8	4	2
6	1	2	5	8	4	9
6	1	2	5	4	8	9
4	1	2	5	6	8	9
2	1	4	5	6	8	9
1	2	4	5	6	8	9

Quick Sort: (1ST pass only)

(5)	0	6	9	7	8	3	2
2	0	6	9	7	8	3	(5)
2	0	(5)	9	7	8	3	6
2	0	3	9	7	8	(5)	6
2	0	3	(5)	7	8	9	6

Mergesort:



3. Trace the search of this array ~~to find~~ '7' using BOTH linear search and binary search. When drawing on the array, label the binary search and label the linear search. /6

Binary

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]
5	6	7	10	11	13	14	15	16	17	19	20	21	23	25

Linear

mid+1 mid-1

For Binary Search:

Low	High*	Mid
0	15	7
0	6	3
0	6	1
2	2	2

* Put Binary on one side and Linear on the other.

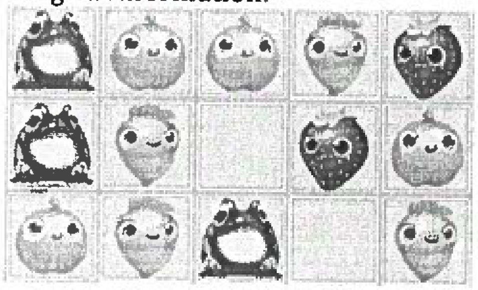
* Remember to start at length not 14.

4. Using the pictures shown, fill in the code for this game of Farm Swap. /5

The picture information:

1 2 3 4 5

The grid information:

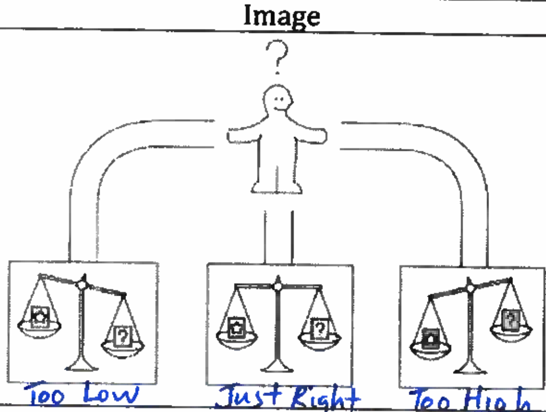
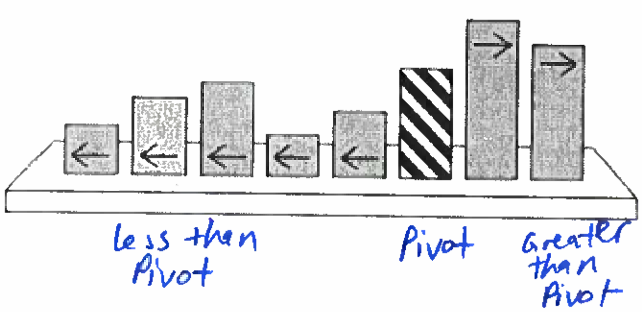


The code:

```
int field [][]={{4, 2, 2, 3, 5},
                {4, 3, 1, 5, 2},
                {2, 3, 4, 1, 3}};

int row = 3;

int col = 5;
JButton pics[]=new JButton[row * col];

Panel grid=new Panel(new GridLayout(row, col));
int m=0;

for(int i=0; i<row; i++){

   for(int j=0; j<col; j++){
       pics[m]=new JButton
        (createImageIcon(field[i][j]+".jpg"));
       pics[m].addActionListener(this);
       pics[m].setActionCommand(m +"");
       grid.add(pics[m]);
       m++;
   }
}
add(grid);
```

Communication

5. Each of the following images were taken from pictorial representations of algorithms. Identify the algorithm shown and explain your choice.
Point form is fine in this section. **Multiple correct answers exist; be careful to explain.** /4

Image	Which Algorithm?	Why that Algorithm?
	Binary Search	In the picture, the person is choosing between scales showing lower, equal and high. This is the same as Binary Search's stopping condition: Is it too high, too low or just right?
	Quick Sort	In the picture, the shaded bar is in it's sorted location, everything before it is less than, everything after it is greater than. This is the same as the end of QuickSort's parition. The pivot is in place as shown in the picture.

6. Provide the phrase or term indicated. /10

Pivot	(a) In quicksort, a partition is finished when this is in its correct place.
John Von Neumann	(b) Wrote Mergesort.
Merge	(c) The second method, besides divide, used by Mergesort.
Bubblesort	(d) The first sorting algorithm that was actually coded. 1963 :)
if(x-1 >=0)	(e) The edge guard for a[x-1][y].
if (y+1 < col)	(f) The edge guard for a[x][y+1].
Quicksort	(g) Tony Hoare won the Turing award for this algorithm in 1980.
number of elements	(h) The n in big-Oh notation stands for this.
swaps	(i) Quicksort does this much more purposefully than Bubblesort.
Recursion	(j) Why an algorithm would have (log n) their algorithm speed. (actually Repeated division)
Searching	(k) Finding the location of an item in an array.

7. Describe the trade-offs for each of the following things. Use one sentence for the positive and another for the negative. /4

(a) Bubble sort

⊕ Positive: When the array is almost sorted, Bubble sort is fastest. It is the only algorithm which can stop early

⊖ Negative: In all other cases, especially random data, bubblesort is slow. It's swaps aren't purposeful + waste time.

(b) Linear Search

⊕ Positive: With unsorted (unordered) data, it is the fastest choice. Binary requires sorted data + sorting is slow.

⊖ Negative: With sorted data, Binary search is much faster; O(log n) for Binary, O(n) for Linear.

8. In the documentary *The Secret Rules of Modern Living*, Marcus du Sautoy points out some key changes of algorithms over time. Fill in this chart to outline the changes he discusses. /6

Single word answers are fine, as long as it is the complete answer.

	Ancient Times	1950s	2000s
At that time, who (or what) wrote the algorithms?	Mathmeticians	Computer Scientists	AI (with human supervision)
At that time, who (or what) were the algorithms written for?	other mathmeticians	computers	Computers
An example of an algorithm developed at that time.	Euclid's GCD	Bubblesort	Kinect Skeletal Tracking System

Thinking

9. Assume that you have a grid that is 9 (rows) x 6 (cols). /4

(a) How many JButtons do you need? 54

(b) Given the JButton's actionCommands, determine each button's (x, y) position in the int tracking array.

4 (0, 4)	36 (6, 0)	43 (7, 1)

10. Which **sorting** algorithm is the best choice for each situation? /5

bubble (a) The list of names is sorted; you add one element to the front. You have extra memory.
selection (b) The array not randomized; it is very large. You just have enough memory to hold it.
merge (c) The char array is in reverse order, but it is only 62 elements long.
quick (d) You have lots of extra memory and the double array is in random order.
quick (e) You have a list of the first 10 million digits of PI. You have no extra memory.

11. In each case, which **search** would be the best choice? /5

binary (a) You have a list of heights, ordered from smallest to greatest.
binary (b) You are looking up a word in a dictionary.
linear (c) You are looking up a definition in a dictionary to find the word that goes with it.
linear (d) You are looking up a student ID in a list with no apparent order.
linear (e) You are looking up a student ID in a list ordered by student last name.

12. Put these algorithm speeds in order.
(1 is fastest, 6 is slowest)

3 O(n)
6 O(n!)
2 O(log n)
1 Constant time
4 O(n log n)
5 $O(n^2)$

13. What speeds are these array algorithms? /8

(a) Swap O(1)
(b) Quicksort O(n log n)
(c) Binary Search O(log n)
(d) Bubble sort (average case) $O(n^2)$
(e) BogoSort O(n!)
(f) Linear Search O(n)

14. Circle **and correct** 4 errors in the binary search method. /4

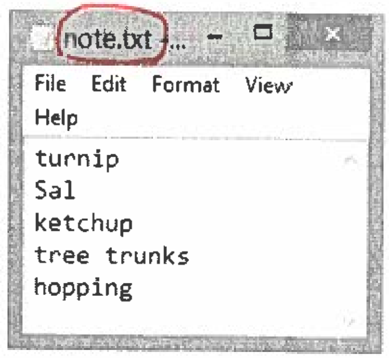
```
public int binarySearch (int a[], int x, int low, int high) {
  if (low > high)
      return -1;

  int mid = (low + high)/2;

  if (a[mid]== x)
       return mid;

  else if (a[mid] < x)
      return binarySearch(a, x, mid+1, high);
  else if
      return binary(a, x, low, mid-1);
}
```

(handwritten corrections: ")" circled after "high"; "==" circled; an "X" written after "(a[mid] < x)"; "if" crossed out; "Search" inserted after "binary")

Application

15. Create a program that reads in information from a text file similar to the one on the right. Then produce the output below, inserting the file's input:

/6



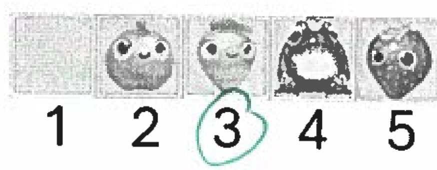

```
I love going to my turnip class. My best friend
Sal is taking turnip with me. Sal
and I both enjoy hopping around turnip class
while writing our detailed notes with ketchup
on tree trunks.
```

```
BufferedReader in;
try {
  in = new BufferedReader (new FileReader (_note.txt_));
  _String_ noun = in.readLine(); turnip
  String friend = in.readLine(); Sal
  String liquid = _in_.readLine(); ketchup
  String surface = in.readLine(); tree trunks
  String verb = in._readLine_(); hopping

  System.out.println("_I love going to my_" + _noun_ + "_class. My best friend_");
  System.out.println(_friend_ +"_is taking_"+ _noun_ +"_with me._"+ _friend_);
  System.out.println("_and I both enjoy_"+ _verb_ +"around"+ _noun_ +"_class_");
  System.out.println("_while writing our detailed notes with_"+ _liquid_);
  System.out.println("_on_"+ _surface_);

  in._close_();
} catch (IOException e) {
   System.out.println ("Error opening file " + e);
}
```

16. Write the java method for the Farm Swap game that assigns a carrot (3) to each element in the field array. Its name is carrotsEverywhere. The variables row and col track the dimensions for the array. Make sure that the new array is updated on the screen using redraw. The method has no parameters, and returns nothing.

/6

```
public _void_ _carrotsEverywhere_ (________)
{
    for (_int i=0; i<row; i++_)
    {
        for (_int j=0; j<col; j++_)
        {
            field [_i_] [_j_] = _3_;
        }
    }
    _redraw_ ();
}
```



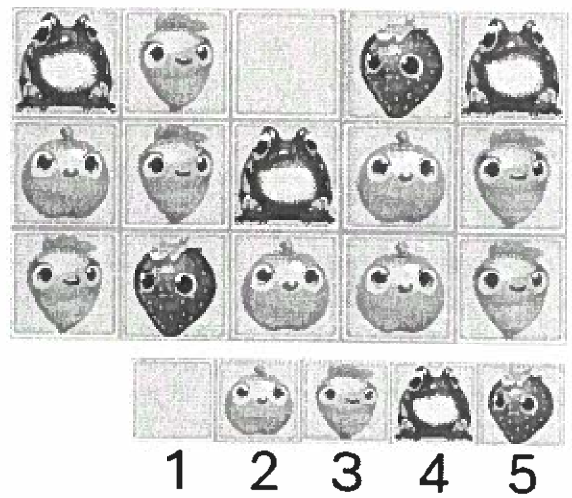
17. Fill in the neighbours chart for the center element. [row][col] /5

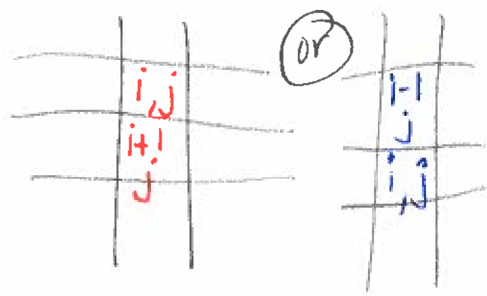
Up-Left	Up	Up-Right
field[i-1][j-1]	field[i-1][j]	field[i-1][j+1]
Left	**Clicked Element**	**Right**
field[i][j-1]	field[i][j]	field[i][j+1]
Down-Left	**Down**	**Down-Right**
field[i+1][j-1]	field[i+1][j]	field[i+1][j+1]

18. In the game Farm Swap, you get extra points every time you get a frog (4) over an apple (2). In the screen shown, the user would get two extra points.

Code a method that returns the number of frogs over apples in a global array named `field`. The variables `row` and `col` track the dimensions for the array.

The method has no parameters but will return an integer.





/7

```
public int frogOverApple()
{
  int pts=0;
  for( int i=0; i<row; i++)
  {
    for (int j=0; j<col; j++)
    {
      if (i+1 < row && field[i][j]==4 && field[i+1][j]==2)
   (or) if (i-1>=0 && field[i-1][j]==4 && field[i][j]==2)
          pts++;
    }
  }
  return pts;
}
```