ICS4U – Review Questions – Unit 4 – Algorithms

1. Speed of swap	O(1)
2. Speed of find length of array	O(1)
3. Speed of find max	O(n)
4. Speed of selection sort	O(n^2)
5. Speed of bubble sort – best case	Close to O(n)
6. Speed of bubble sort – average case	O(n^2)
7. Speed of quick sort	O(n log n)
8. Speed of mergesort	O(n log n)
9. Speed of merge	O(n)
10. Speed of binary search	O(log n)
11. Speed of linear search	O(n)
12. Speed of bogosort	O(n!)
13. In Big-Oh notation, the O stands for…	Order
14. In Big-Oh notation, the n stands for..	Number of elements in the array
15. Which is faster: binary or linear search	Binary
16. Which is faster: searching or sorting	Searching
17. Which is faster: quicksort or mergesort?	Quicksort
18. Which is faster: bubblesort (average) or selection sort?	Selection Sort
19. Put the 4 sorts in order, fastest to slowest	Bubble (best case), Quick, Merge, Selection
20. Why is selection sort called selection sort?	SELECTING best element (max)
21. Why is quicksort called quicksort?	It is the QUICKEST. Purposeful swaps.
22. Why is mergesort called mergesort?	It divides the array into sorted arrays and MERGES them together
23. Why is bubblesort called bubblesort?	The billions of little swaps are like bubbles rising in pop.

24. Why is binary search called binary search?	It splits the remaining part of the array in half. Halfing is based on 2 or BINARY
25. Why is linear search called linear search?	It moves in a LINE down the array
26. What is the hidden cost of binary search?	Data must be sorted AND sorting is a slow operation
27. Why don't we measure algorithm speed in terms of seconds or time?	It is hardware dependent. Big-Oh is based on the algorithm.
28. Term for putting elements in order	Sorting
29. Term for finding an element in an array	Searching
30. Good hardware cannot compensate for ….	A bad algorithm
31. Term for a series of steps that complete a task	Algorithm
32. What is the fastest in-place algorithm in the general case?	Quicksort
33. What is Quicksort's title?	Fastest in-place algorithm in the general case.
34. Where is the pivot located at the start of a Quicksort partition?	At the beginning of the array
35. Where is the pivot located at the end of a Quicksort partition?	It is in its correct location.
36. At the end of a quicksort partition, what is on the left of the pivot?	Elements smaller than the pivot
37. At the end of a quicksort partition, what is on the right of the pivot?	Elements larger than the pivot
38. Which sorting algorithm is not "in-place"?	Mergesort
39. What does "in-place" algorithm mean?	It uses swaps. It doesn't need extra memory.

40. Which two sorting algorithms are recursive?	Quicksort and mergesort
41. What are the two parts of mergesort?	Divide and merge
42. What is the slowest sorting algorithm?	Bogosort
43. Why is quicksort better than bubblesort?	More purposeful swaps. Moves to correct half of array.
44. How do you know the element isn't in the array in binary search?	High < Low **or** Low > High
45. Who invented Quicksort?	Tony Hoare
46. Who invented Mergsort?	John Von Neumaan
47. Who was a pioneer in AI research, developed the first computer and helped the allies win WWII?	Alan Turing
48. Who made the documentary Secret rules of Modern living?	Marcus du Satouy
49. Who wrote down the first algorithm?	Euclid
50. What was the first algorithm written down?	GCD (greatest common divisor)
51. Who wrote that good hardware cannot compensate for a slow algorithm?	Jon Bentley
52. (Secret Rules Documentary) What group developed the first algorithms?	Mathematicians
53. (Secret Rules Documentary) What group developed in the 1960s?	Programmers
54. (Secret Rules Documentary) What group develops algorithms now?	AI (Machine Learning)
55. (Secret Rules Documentary) Who were the first algorithms written for?	Mathematicians
56. (Secret Rules Documentary) Who were the algorithms written for in the 1960s?	Computers
57. (Secret Rules Documentary) Who are algorithms written for now?	Humans

58. First sorting algorithm CODED	Bubble sort
59. First sorting algorithm CREATED	Mergesort
60. What is the first test to determine which sorting algorithm to use?	Almost sorted. Use Bubble.
61. What is the second test to determine which sorting algorithm to use?	Random order. Use Quick.
62. What is the third test to determine which sorting algorithm to use?	Enough memory. Use Merge
63. What is the first test to determine which searching algorithm to use?	Sorted? Use Binary Not Sorted? Use Linear
64. Put the sorting speeds in order, fastest to slowest	O(1), O(log n), O(n), O(n log n), O(n^2), O(n^3), O(n!)
65. Positive of Quicksort	Really fast. O(n log n)
66. Negative of Quicksort	Complex. Only for random data. Reverse order **or** Almost sorted = bad
67. Positive of Mergesort	Really fast. O(n log n)
68. Negative of Mergesort	Requires extra memory
69. Positive of Bubble sort	If almost sorted, close to O(n). That's fast
70. Negative of Bubble sort	In all other cases, slow O(n^2). A lot of swaps.
71. Positive of Selection sort	Easy to understand. Based on max.
72. Negative of Selection sort	Slow. Simplicity isn't efficient.
73. Positive of Binary search	Fast. Really fast. O(log n)
74. Negative of Binary search	Requires sorted data. Sorting is slow.
75. Positive of Linear search	Works even for unsorted data

76. Negative of Linear search	Slower than binary search.
77. What is the edge guard for i-1?	i-1 >=0
78. What is the edge guard for i+1?	i+1<row
79. What is the edge guard for j-1?	j-1>=0
80. What is the edge guard for j+1?	j+1<col
81. What is the outer for loop for a coding question?	for(int i=0; i<row; i++)
82. What is the inner for loop for a coding question?	for(int j=0; j<col; j++)
83. Which way is the row?	Horizontal (i)
84. Which way is the column?	Vertical (j)
85. If the actionCommand is n, what is the row?	n/col
86. If the actionCommand is n, what is the column?	n%col