What is a stable sort? _____
1) The sort uses the paradigm of a barn's stable to partition data into different stalls similar to shunting-yard alg.
2) Duplicate values are in the same order after the sort as they were in the original data
3) Duplicate values are eliminated so the sorted data has no duplicates
4) The sort is done in place without using additional space or data structures
5) The sort is always the same runtime complexity no matter what order the original data is in

Which version of Fib(n) will be more space efficient for calculating Fib(80) after first calculating Fib(95)?
_____ 1) Iterative 2) Recursive 3) Dynamic Programming 4) All are equal

What is the main reason why? _____ (Use the numbers below to answer this question)
1) recursive version takes advantage of two base cases
2) iterative and dynamic programming versions call the recursive version so they are all equal
3) all versions have the same big-Oh time and space complexity
4) all versions have the same recurrence relation and closed-form solution
5) recursive version is shorter to code
6) recursive version implicitly uses the run time stack
7) iterative version does not re-compute all the previously computed fib values each time it is called
8) iterative version uses looping and nothing else
9) dynamic programming uses dynamic memory allocation to calculate fib values
10) dynamic programming version stores previously computed fib values so does not re-compute them

Which version of Fib(n) will be more time efficient for calculating Fib(80) after first calculating Fib(95)?
_____ 1) Iterative 2) Recursive 3) Dynamic Programming 4) All are equal

What is the main reason why? _____ (Use the numbers above to answer this question)

If you have an algorithm that when you double the number of elements it …

doubles the number of comparisons, this algorithm is most likely in what big-Oh complexity class? ______

1) \( O( n \log_2 n ) \)
2) \( O( n! ) \)
3) \( O( 1 ) \)
4) \( O( \log_2 n ) \)
5) \( O( 2^n ) \)
6) \( O( n^2 ) \)
7) \( O( n ) \)

squares the number of comparisons, this algorithm is most likely in what big-Oh complexity class? ______

1) \( O( n \log_2 n ) \)
2) \( O( n! ) \)
3) \( O( 1 ) \)
4) \( O( \log_2 n ) \)
5) \( O( 2^n ) \)
6) \( O( n^2 ) \)
7) \( O( n ) \)

does not change the number of comparisons, this alg. is most likely in what big-Oh complexity class? ______

1) \( O( n \log_2 n ) \)
2) \( O( n! ) \)
3) \( O( 1 ) \)
4) \( O( \log_2 n ) \)
5) \( O( 2^n ) \)
6) \( O( n^2 ) \)
7) \( O( n ) \)

increases the number of comparisons by 1, this alg. is most likely in what big-Oh complexity class? ______
Given the binary tree to the right

Specify the output for the following traversals

Preorder traversal: ____________________________

Inorder traversal: ____________________________

Postorder traversal: ____________________________

Breadth-first traversal: ____________________________

What is Rick's favorite algorithm? ____________________________

Construct a minimum spanning tree from the following network. Hint: 9 vertices so mst should have 8 edges.

What is the total weight of the minimum spanning tree? ______

Is there more than one minimum spanning tree in this graph (yes or no)? ______

Use the following numbers for the next series of questions.

1) Dennis Ritchie  
2) Brian Kernighan  
3) Ken Thompson

_____Coauthor of the AWK programming language?

_____The father of the C programming language?

_____The father of grep?

_____Coined the term Unix?