CSIS385 Algorithms

Lab #10: Limitations of Algorithm Power

Science is what we understand well enough to explain to a computer. Art is everything else we do.

Donald Knuth.

Names: _____

Learning goals:

- To understand binary decision trees
- To be able to draw binary decision trees for standard algorithms
- To be able to derive lower bounds using binary decision trees

1) You created a decision tree for playing the game Guess Who with 9 characters. In the worst case, how many questions will it ask? _____ Do you think it asks the fewest possible questions in the worst case, or do you think you could design a better tree that asks fewer questions in the worst case?

2) How is the height of your decision tree related to the worst case number of questions asked? In other words, if your decision tree has height h, what is the worst case number of questions you will have to ask?

3) How many leaf nodes does your Guess Who decision tree have? _____

4) Could you have designed a Guess Who decision tree for 9 characters using fewer leaves? Why or why not?

5) If you have a binary decision tree of height 3, what is the maximum possible number of leaf nodes in the tree? _____

6) If you have a binary decision tree of height 4, what is the maximum possible number of leaf nodes in the tree? _____

7) What is the smallest possible height for a Guess Who decision tree with 9 characters? _____ Clearly argue why it cannot be lower than this based on the number of leaves any such tree must have.

8) If you have a binary decision tree of height h, what is the maximum possible number of leaf nodes in the tree? _____

9) Suppose your Guess Who game had more characters to choose from. For each of the number of characters below, give the minimum possible height of a decision tree that plays the game? Fill in each row.

Total number of characters to choose from in the game	Minimum number of leaf nodes in the decision tree	Minimum possible height of the decision tree
31		
32		
33		
63		
64		
65		
n		

10) Consider the problem of finding the location of a value X among an **ordered** sequence of nine values (a, b, c, d, e, f, g, h, i). Assume you are guaranteed that X is one of these items. Draw two binary decisions trees for solving this problem, one corresponding to a sequential search strategy and the other corresponding to a binary search strategy. Internal nodes should contain comparisons of the form "a < b" or "a > b". Each leaf node should contain an answer, which is the index of the found item.

Sequential Search Decision Tree:

Worst case number of comparisons made using your sequential search decision tree:_____

Binary Search Decision Tree:

Worst case number of comparisons made using your binary search decision tree:_____

11) How is the height of your decision tree in problem #10 related to the number of comparisons the algorithm makes to find the answer in the worst case? In other words, if the height of the tree is h, the number of comparison operations performed by the algorithm in the worst case is at least _____.

12) What is the smallest possible height of ANY decision tree solving problem #10 with 9 ordered values? Clearly argue why it cannot be lower than this, based on the number of leaves any such tree must have.

13) Suppose you have n ordered values instead of just the 9 values in problem #10. What is the smallest possible height of ANY decision tree solving this problem with n elements? Give a precise formula. Clearly argue why it cannot be lower than this, based on the number of leaves any such tree must have.

14) EVERY algorithm that uses comparisons to find X in an ordered sequence can be represented by a decision tree, just as you did for sequential and binary search. So suppose someone says they have a fast new comparison based algorithm solving this problem that makes fewer comparisons than the formula you gave in #13. Clearly argue why this is impossible.

15) Consider the "sorting game" in which you want to guess the correct sorted ordering of 3 unique values (a, b, c) whose actual values you don't know. Design a decision tree for playing this game and draw it below. Internal nodes contain comparisons of the form "a < c" or "b > a". Leaf nodes contain the different orderings of (a, b, c).

16) In the worst case, how many comparisons does your decision tree make in determining the correct ordering of (a, b, c)? _____

17) Does your decision tree have minimum possible height? _____

18) If you play the sorting game using n=4 values (a, b, c, d), then you will need at least _____ leaf nodes in your decision tree.

19) The minimum possible height for a decision tree that plays the sorting game with n=4 is _____.

20) If you play the sorting game using n=5 values (a, b, c, d, e), then you will need at least _____ leaf nodes in your decision tree.

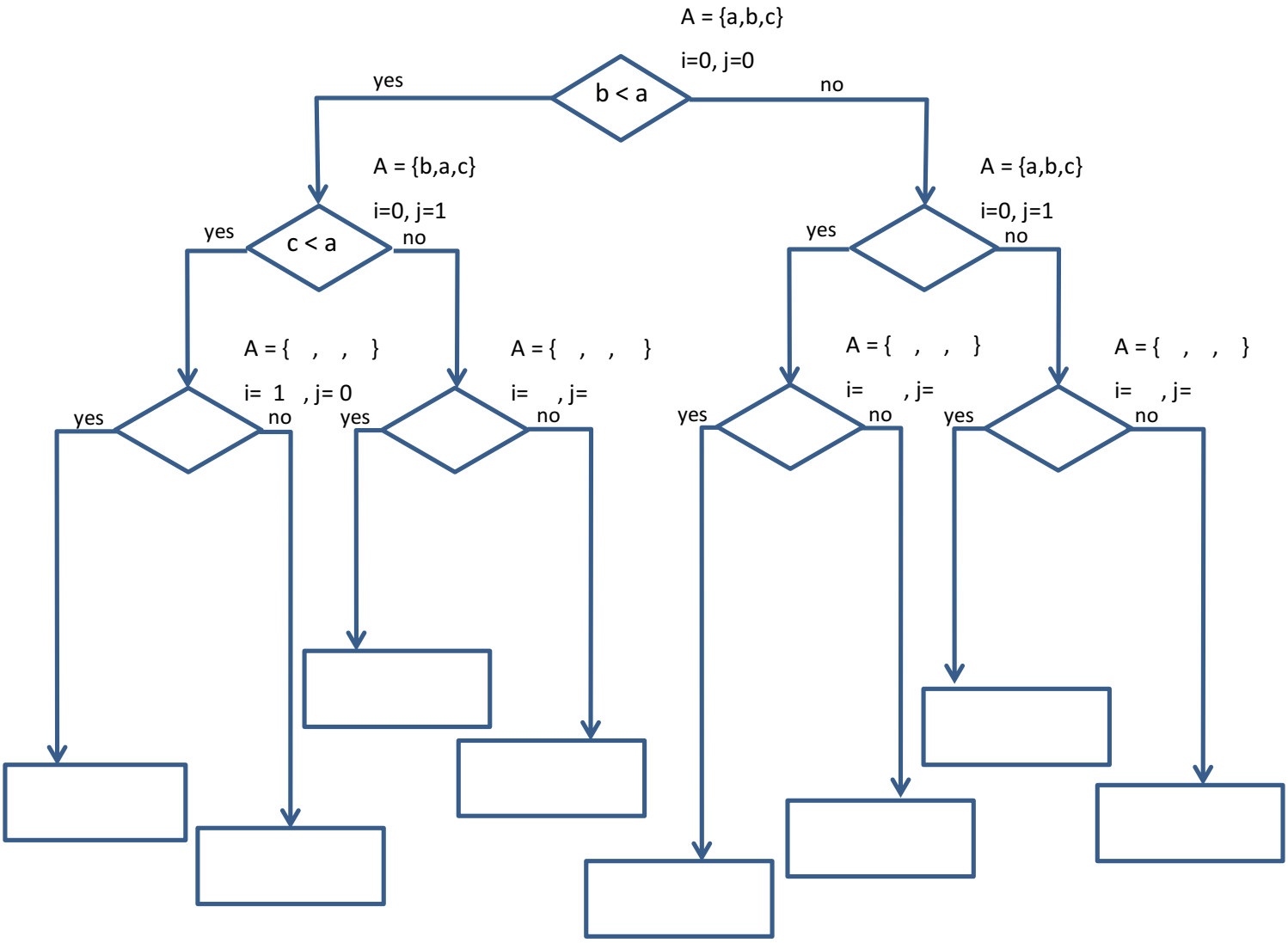
21) The minimum possible height for a decision tree that plays the sorting game with n=5 is _____.

22) In general, if you play the sorting game using n values, then you will need at least _____ leaf nodes in your decision tree.

23) Therefore the height of ANY decision tree for the sorting game using n values must be at least _____.

24) EVERY sorting algorithm that uses comparisons has a corresponding binary decision tree that represents the different sequences of comparisons that the algorithm makes as it sorts the values. For example, complete the binary decision tree for sorting values A[0…2] = {a, b, c} using the BubbleSort1 algorithm below. Two nodes are filled in for you already. Next to each node we noted the order of values a, b, c in A after the swap (if any) is done based on the result of the comparison, along with the current values of i and j. Fill in the remaining parts of the decision tree in a similar way. The internal nodes should display the next comparison made, and the leaf nodes should display the final sorted orderings.

```
BubbleSort1( A[0…n-1] )
    for i ← 0 to n - 2
        for j ← 0 to n – 2 - i
            if  A[ j+1 ] < A[ j ]
                swap A[ j+1 ] and A[ j ]
```



25) Because EVERY comparison based sorting algorithm has a corresponding decision tree, and because the height of ALL decision trees for sorting n values is at least log(n!), we can conclude that the number of comparison operations performed by EVERY comparison based sorting algorithm in the worst case is at least _____.