|  |  |
| --- | --- |
|  | **Homework #2** Sidney Sanders |
| CS 4120 | Due: Wed, Sept. 29, 2021 |
| Fall, 2021 | 30 points |

1. Exercise 4.3-2, page 87 of your text.

Show that the solution of *T* (*n*) = *T* ( *n*/2 ) + 1 is *O*(lg *n*).

T(n) = T(n/2) + 1

≤ 3 log (n/2) -1 +1

≤ 3 log (3n/4)

= 3 log n + 3 log (3/4)

≤ 3 log n + log (1/2)

=3 log n - 1

=O (log *n*)

~~2~~. What is the time complexity *T*(*n*) of the nested loops below? For simplicity, you may assume

that *n* is a power of 2. That is, *n* = 2*k* for some positive integer *k*.

:

i = n;

**while** (i >= 1){

j = i;

**while** (j <= n) {

<body of the inner while loop > // Needs (1).

j = 2 \* j;

}

i = i/2 ;

}

:

The inner loop is log(n) and the outer loop is log(n)

Making it log(n) \*log(n) = log^2(n) = Olog(n)

1. Use recursive binary search to search for the integer 100 in the following list (array) of integers. Show the actions step by step.

12 34 37 45 57 82 95 100 134

~~12 34 37 45 57~~ 82 95 100 134

82 95 100 134

~~82 95~~ 100 134

100 134 (found)

4. ~~a.~~ Use Merge sort to sort the following list. Show the action step by step.

123 34 189 56 150 12 9 240

123 34 189 56 | 150 12 9 240

123 34 | 189 56 | 150 12 | 9 240

123 | 34 | 189 | 56 | 150 | 12 | 9 | 240

34 123 | 56 189 | 12 150 | 9 240

34 56 123 189 | 9 12 150 240

9 12 34 56 123 150 189 240

* 1. Give the tree of recursive calls in 4-a.

123 34 189 56 150 12 9 240

150 12 9 240

123 34 189 56

150 12

9 240

189 56

123 34

240

9

12

150

56

189

34

123

1. a. Suppose that, we were to search a list of 100,000 items using binary search. What is the maximum number of comparisons that this algorithm must perform before finding a given item or concluding that it is not in the list?
   1. In general, if you were to perform a “ternary search” (by dividing the list into three approximately equal parts and eliminating two of the three thirds at “each level”), do you think you can obtain a better θ (theta) efficiency? Explain your rationale.
2. To find the number of comparisons log2100,000 = 16.6 => 17 comparisons to find the given item, worst case.
3. Ternary will give better result because with three divisions, the list is split better.

Log3100,000 = 10.5 => 11 comparisons worst case. This is quicker and would take less time to execute for large data collections.

1. ~~[FOR CS 5120 ONLY; CS 4120 students may use this problem for practice but are not required to submit.]~~

~~Write an algorithm that searches a sorted list of~~ *~~n~~* ~~items by dividing it into three sublists of almost~~ *~~n~~*~~/3 items. This algorithm finds the sublist that might contain the given item and divides it into three~~~~smaller sublists of almost equal size. The algorithm repeats this process until it finds the item or concludes that the item is not in the list. Analyze your algorithm, and give the results using order notation.~~