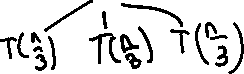
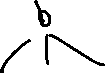
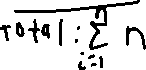
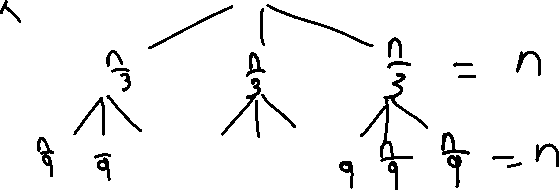
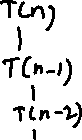
**Algorithms homework related to exam 1, Summer 202**

# Hw2: Due Monday, June 7, at 11:59pm

1. .
   1. Use a recursion tree to show that *S*(*n*) = Θ(*g*(*n*)), for some function *g*(*n*).
   2. Use induction to prove that *S*(*n*) = Ω(*g*(*n*)).
2. Recursion tree



1. Induction:
2. *T*(*n*) = *T*(*n*−1) + log *n*. Assume that there is an appropriate base case so that you don’t get an undefined function.
   1. Draw a recursion tree for *T*(*n*). Identify its depth and the amount of work per level. Use “exaggerate and simplify” to get an upper bound for *T*(*n*), and then underestimate and simplify to get a matching lower bound. (Always using Theta notation). Your bound should be in a form that is easily recognizable (and easy to compare to simple functions like linear, quadratic, etc).

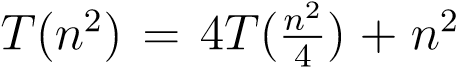


* 1. Derive the same upper bound for *T*(*n*), by induction.

1. Solve *T*(*n*) = *T*(√*n*) + 1, using a change of variables: *n* = 2*m*.

*To warm up, consider the following question. (Don’t submit an answer for the warmup)* Think of an abstract recursive algorithm that operates on a *n* × *n* matrix: It does non-recursive work proportional to the number of elements in the matrix (say just *n*2 work), and then recurses on each of the four quadrants. Assume that quadrants are nicely defined.

This could be expressed as , meaning the parameter used in S to describe the problem is the matrix side length.

But suppose that we hadn’t thought of doing that, and instead we expressed the time complexity as. Here, the parameter used for the problem size is the total number of elements. In this case, because this parameter in T is a function of *n* (other than just *n* itself), we don’t really have a direct method to deal with it. But we could set *m* = *n*2, so .

What’s the solution in each case (in terms of n)? It should be the same.

Besides that, the lesson here is that there can be multiple ways to quantify a problem size and recurse accordingly, e.g. *S*(*n*) vs *T*(*n*2). You’ll need to consider this after you apply the change of variables as suggested above for the actual homework problem.

4. Use the master method for the following (state case or explain what dominates, and state the answer), or explain why it’s not possible.

