

CSE 202 Homework 1
Winter, 2018
Due Wed. Jan 17 at 4 PM

Exercises

Which of the following pairs of functions are of the same order?
Explain your answer

1. $n \log(n^2)$ and $n \log n$
2. $2^{\log n}$ and $2^{2 \log n}$.
3. 2^n and $2^n + 2^{n-1} + 2^{n-2} + \dots + 1$
4. n^2 and $n^2 + (n-1)^2 + (n-2)^2 + \dots + 4 + 1$
5. $n \log n$ and $\log(n!)$

The Fibonacci sequence Fib_i is defined by the recurrence $Fib_0 = Fib_1 = 1$, $Fib_{i+1} = Fib_i + Fib_{i-1}$ for $i \geq 1$.

Give a simplified expression for $Fib_1 + Fib_3 + Fib_5 + \dots + Fib_{2i+1}$, and prove your answer.

You start with some number of red balls in a red urn, and some number of green balls in a green urn. At each step, you pick a ball arbitrarily from each urn, and switch both to the other urn. Prove that, at all times, there are the same number of red balls in the green urn as there are green balls in the red urn.

Problems Each problem is worth 20 points.

Recurrence Let $T(n)$ be the function given by the recursion: $T(n) = nT(\lfloor \sqrt{n} \rfloor)$ for $n > 1$ and $T(1) = 1$. Is $T(n) \in O(n^k)$ for some constant k , i.e. is T bounded by a polynomial in n ? Prove your answer either way. (Note: logic and definition of O notation are more important than exact calculations for this problem.)

Reasoning about order Let $f(n)$ be a positive, integer-valued function on the natural numbers that is non-decreasing. Show that if $f(2n) \in O(f(n))$, then $f(n) \in O(n^k)$ for some constant k . Is the converse also always true (Note: this is the difficult part)?

Triangles: A triangle in a graph are 3 nodes any two of which are adjacent. Present two algorithms for determining whether a graph has a triangle, one if the graph is given as an adjacency matrix and the other if it is given in adjacency list format. Analyze these algorithms in terms of both the number of nodes n and the number of edges m .

Base Conversion Present and analyze an $O(n^2)$ time algorithm that inputs an array of n base 10 digits representing a positive integer in base 10 and outputs an array of base 2 bits representing the same integer in base 2. Count each operation on a single digit as a step, e.g., adding two n bit binary strings takes time $O(n)$ since one addition involves $O(n)$ bit operations. Be sure to prove that your algorithm is correct.

Experimental Evaluation of Triangle Algorithm . Implement your triangle finding algorithm the above algorithm, and test it on many random graphs where each edge is present with probability $1/2$. Try it for n as many different powers of 2 as you can. Plot time vs. input size on a log vs. log curve. Does the algorithm's observed time fit the analysis? Why or why not? Then do the same experiment for random bipartite graphs with n vertices on each side.