CSE 202 Homework 1
Winter, 2022
Due Wed., Jan 12 at 11:59 PM
Background: order notation, induction , loop invariants, simple analysis of algorithms

Exercises

Which of the following pairs of functions are of the same order? (By same order, I mean that they have the Θ relation.) 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 $2^n + 2^{n-1} + 2^{n-2} + ...1$
4. n^2 and $n^2 + (n-1)^2 + (n-2)^2 + ...4 + 1$
5. $n \log n$ and $\log(n!)$

The Fibonnaci 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 expresssion for $Fib_1 + Fib_3 + Fib_5 + ...Fib_{2i+1}$, 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.

Ungraded Problems

1. 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.)

2. 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)?

3. Consider the following word puzzle. You are given a vocabulary list of n, k-letter words, a starting k-letter word, and an ending k-letter word. You want to find a way to change the starting word to the ending word one letter at a time, always being on the list of vocabulary words. Define

a graph where solutions correspond to certain paths in the graph. Be sure to specify what the vertices and edges of your graph are, and exactly which paths correspond to solutions. For example, say $k = 4$ and $n = 15$, the starting word is $SHOW$ and the ending word is $TELL$, and the list contains those two words and

$$SLOW, STOW, STOP, SHOP, SLOP, CHOP$$

,

$$CROP, DROP, CLOP, COOP, COOL, TOOL, TOLL$$

. Then one solution is

$$SHOW, SLOW, SLOP, CLOP, COOP, COOL, TOOL, TOLL, TELL$$

Graded Problems - 20 points each

1. Prove that, if for positive integer-valued functions f, g, $f \in \Theta(g)$, and h is a strictly increasing positive integer valued function, that $f(h(n)) \in \Theta(g(h(n))$. Is the converse always true?

2. One vector of real numbers $(x_1, ..x_d)$ dominates another vector $(y_1, ..y_d)$ if $x_i \geq y_i$ for $i = 1...d$. (Think of each co-ordinate as a quantity that might be better or worse for some alternatives, e.g., speed and memory for computer architectures. $\vec{x}$ dominates $\vec{y}$ if it is at least as good in all respects.) Given a set of n vectors in d dimensions, S , $\vec{x}$ is Pareto Optimal if no $\vec{y} \in S, \vec{y} \neq \vec{x}$, dominates $\vec{x}$. For two dimensions, give an efficient algorithm that returns the list of Pareto Optimal elements of S. (Extra: how about for larger values of d?)

3. Triangles: A triangle in a graph are 3 nodes any two of which are adjacent. Present two algorithms for determining whether a simple undirected graph has a triangle, if the graph is given as an adjacency matrix. Analyze this algorithm in terms of both the number of nodes n and the number of edges m.

4. 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.