COMPSCI 250: Fall 2023
Homework 4

David A. Mix Barrington , Ghazaleh Parvini

Due Date : Friday, November 3

This assignment has 9 problems. There is also 1 Extra Credit problem. The extra credit is 10 points.

Please submit a single PDF file, with the problems in order (as below), and legible. Look at your PDF before submitting it – it is fine to scan or photograph a handwritten document but it the graders can't read it, they won't grade it.

Please assign pages to problems in Gradescope. Graders will click on the problem number. If no page shows up because it's not assigned, the assumption is you have not solved the problem.

Be sure you are doing Problems in the book and not Exercises: the numbers should start with P rather than E.

For full credit, show your work, explaining your reasoning. This also helps assign partial credit.

You are responsible for following the academic honesty guidelines on the Grading and Requirements page. This means that what you present must be your own work in presentation, and you must acknowledge all sources of aid other than course staff and the textbook. You will get 2 extra points if you typeset your Homework.

(12 points) **Problem 4.4.7**

A polygon is called **convex** if every line segment from one vertex to another lies entirely within the polygon. To **triangulate** a polygon, we take some of these line segments, which don't cross one another, and use them to divide the polygon into triangles. Prove, by strong induction for all naturals n with $n \geq 3$, that every convex polygon with n sides has a triangulation, and that every triangulation contains exactly $n - 2$ triangles. (**Hint:** When you divide an n-gon with a single line segment, you create an i-gon and a j-gon for some naturals i and j. What does your strong inductive hypothesis tell you about triangulations of these polygons?)

(12 points) **Problem 4.7.6**

(uses Java) We can define the **balanced parenthesis language** using recursion. This is the set of sequences of left and right parentheses that are balanced, in that every left paren has a matching right paren and the pairs are nested properly. We'll use "L" and "R" instead of "(" and ")" for readability.

We define the language Paren by the following four rules:

(a) λ is in Paren.

(b) If u is in Paren, then so is LuR.

(c) If u and v are in Paren, then so is uv.

(d) No other strings are in Paren

Write a real-Java static method (or a Python function) `isBalanced` that takes a `String` argument and returns a `boolean` telling whether the input string is in Paren. A non-recursive method is simpler.

(10 points) **Problem 4.9.2**

Prove that any directed cycle in the graph of a partial order must only involve one node. (**Hint:** If the cycle were to contain two distinct nodes x and y, what does transitivity tell you about arcs between x and y?)

(10 points) **Problem 4.10.5**

Prove that if T is any rooted directed binary tree (where every internal node has out-degree exactly two), then the number of leaves in T is one greater than the number of internal nodes. (**Hint:** Use induction on the definition of such trees.)

(12 points) **Problem 4.11.1**

Show that a $3 \times n$ rectangle can be covered exactly with L-shaped tiles if and only if n is even. (**Hint:** For the negative result, use induction on all odd numbers and an indirect proof in the inductive step.)

(12 points) **Problem 4.11.4**

Prove the claim at the end of the section about the Euclidean Algorithm and Fibonacci numbers. Specifically, prove that if positive naturals a and b are each at most $F(n)$, then the Euclidean Algorithm performs at most $n - 2$ divisions. (You may assume that $n > 2$.) (It follows from this result that Fibonacci numbers are the worst case, but you may not use that fact to solve this problem!)

(10 points) **Problem 9.1.5**

State and prove a theorem giving the maximum and minimum possible number of leaves in a rooted tree of depth d and degree k. Repeat for the maximum and minimum total number of nodes.

(10 points) **Problem 9.3.5**

Let T be a parse tree for an expression with only unary and binary operators, and let m be the number of primitive elements in the expression. Prove that if $m > 1$, then there exists a node x of T such that the subtree rooted at x contains at least $m/3$ of the primitive elements and at most $2m/3$ of the primitive elements.

(12 points) **Problem 9.4.7**

Following Exercise 9.4.9, describe the state graph of the Towers of Hanoi puzzle for general n. Prove that the puzzle is always solvable, and find the number of moves in the shortest possible solution. (It will be useful to have a recursive definition of the state space.)

Extra Credit

(10 points) **Problem 9.5.8**

In the knight's tour problem, we are looking for a path of knight's moves from a square of the $n \times n$ chessboard to itself, such that the path visits all n^2 squares of the board. How can we modify BFS or DFS to solve this problem? (The solution may not be efficient, but it is correct.)