CPSC 031 — Mathematics Review for CPSC 413

Exercise #1 — Mathematical Induction

September, 2000

Please try these exercises before the 6pm lecture on September 6.

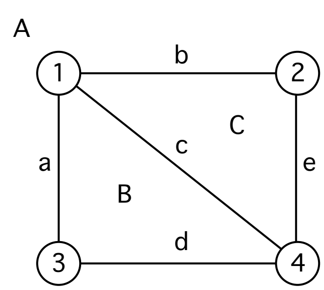
You might find that these exercises are quite challenging! If they are, and you would like to start with some exercises that are bit simpler, then you should begin with the "warmup problems" instead.

It may help, for these questions, if you experiment a bit with the data structure, formulas, or algorithm mentioned in the problem. For example, if a question asks about regions in the plane separated by n lines, it might help to draw pictures that include 0 lines, 1 line, 2 lines, and so on, and look for a pattern that will help you to get started. Similarly, if a question asks something about an algorithm, then it's often useful to start by tracing execution of the algorithm by hand on one or more small inputs.

1. A *planar graph* is a graph that can be drawn (on a planar surface, like a blackboard) so that no two edges intersect except at endpoints. A graph is *connected* if there is a path (along edges) between any pair of distinct vertices in the graph.

   Prove *Euler's formula*: If G is a nonempty connected planar graph with v vertices, e edges, and f faces, then $v + f = e + 2$.

   For example, the connected planar graph shown below has $v = 4$ vertices, numbered 1–4. It has $e = 5$ edges, labelled a–e. It has $f = 3$ faces (including an exterior face) labelled A–C. In this case, $v + f = 7 = e + 2$, as expected.



2. A node of a binary tree is an *internal node* if it has at least one child, and it is a *leaf* otherwise. A binary tree is a *complete* binary tree if every internal node has exactly two children.

   Prove that every nonempty complete binary tree has exactly one more leaf than it has internal nodes.

3. Lines in the plane are in *general position* if no two lines are parallel and no three (or more) lines intersect at a common point. Find the number $R(n)$ of regions created by n lines in general position, and prove your answer.

4. Consider the algorithm, *Convert-to-Binary*, that is shown below.

   Algorithm *Convert-to-Binary*

   Input: A positive integer n.

   Output: An array B of bits corresponding to the binary representation of n.

   ```
   begin
     t := n
     k := 0
     while t > 0 do
       k := k + 1
       B[k] := t mod 2
       t := t div 2
     end while
   end
   ```

   Note that (a **mod** b) and (a **div** b) are, respectively, the remainder and quotient you would get when you divide a positive integer a by a positive integer b, so that (a **mod** b) is always between 0 and $b-1$, inclusive, and so that it is always true that

   $$a = (a \ \mathbf{div} \ b) \times b + (a \ \mathbf{mod} \ b).$$

   Prove that when the above algorithm terminates, the binary representation of the integer n is stored in the array B.

5. Find a positive constant c such that the following is true for every positive integer n and prove your result.

   $$\sum_{i=1}^{n} i^2 \leq cn^3.$$

6. Once again, consider the above algorithm *Convert-to-Binary*. Find positive constants c and d such that, given any positive integer n as input, the number of statements executed by the algorithm is at most $c \log_2 n + d$. Prove that your choices are correct.