COMP 2711 Discrete Mathematical Tools for Computer Science
2022 Fall Semester – Tutorial 11

Question 1:

(a) Find a recurrence relation for the number of bit strings of length n that contain a pair of consecutive 0s.

(b) What are the initial conditions?

(c) How many bit strings of length seven contain two consecutive 0s?

Solution :

(a) Let a_n be the number of bit strings of length n containing a pair of consecutive 0's. In order to construct a bit string of length n containing a pair of consecutive 0's we could start with 1 and follow with a string of length $n-1$ containing a pair of consecutive 0's, or we could start with a 01 and follow with a string of length $n-2$ containing a pair of consecutive 0's, or we could start with a 00 and follow with any string of length $n-2$. These three cases are mutually exclusive and exhaust the possibilities for how the string might start. From this analysis we can immediately write down the recurrence relation, valid for all $n \geq 2$: $a_n = a_{n-1} + a_{n-2} + 2^{n-2}$. (Recall that there are 2^k bit strings of length k .)

(b) There are no bit strings of length 0 or 1 containing a pair of consecutive 0's, so the initial conditions are $a_0 = a_1 = 0$.

(c) We will compute a_2 through a_7 using the recurrence relation:

$$\begin{aligned}
a_2 &= a_1 + a_0 + 2^0 = 0 + 0 + 1 = 1 \\
a_3 &= a_2 + a_1 + 2^1 = 1 + 0 + 2 = 3 \\
a_4 &= a_3 + a_2 + 2^2 = 3 + 1 + 4 = 8 \\
a_5 &= a_4 + a_3 + 2^3 = 8 + 3 + 8 = 19 \\
a_6 &= a_5 + a_4 + 2^4 = 19 + 8 + 16 = 43 \\
a_7 &= a_6 + a_5 + 2^5 = 43 + 19 + 32 = 94
\end{aligned}$$

Thus there are 94 bit strings of length 7 containing two consecutive 0's.

Question 2: You are given a real number a and a positive integer n that is a power of 2, i.e. $n = 2^k$ for some integer $k \geq 0$.

(a) Devise a recursive algorithm to find a^n. Your algorithm should use as few multiplications as possible.

(b) Give a recurrence equation of the number of multiplications used in your algorithm in (a).

(c) Solve your recurrence equation in (b).

Note that in this question, we assume

(i) b^m uses $m-1$ multiplications for any real number b and positive integer m.

(ii) b/m is not counted as a multiplication if m is a positive integer.

(iii) In the computation of $(f(n))^m$, $f(n)$ is evaluated only once. But, in the computation of $f(n) \cdot f(n)$, $f(n)$ is evaluated twice.

Solution : (a) **procedure** *twopower*(n: positive integer, a: real number)
if $n = 1$ **then return** a
else return $(twopower(n/2, a))^2$

(b) $T(1) = 0$. $T(2^k) = T(2^{k-1}) + 1$ for $n > 1$.

(c)

$$\begin{aligned} T(2^k) &= T(2^{k-1}) + 1 \\ &= T(2^{k-2}) + 2 \\ &\vdots \\ &= T(2^0) + k \\ &= k \\ &= \log_2 n \end{aligned}$$

Question 3: The **reversal** of a string is the string consisting of the symbols of the string in reverse order. The reversal of the string w is denoted by w^R.

(a) Give a recursive definition of the reversal of a string. [*Hint*: First define the reversal of the empty string. Then write a string w of length $n+1$ as xy, where x is a string of length n, and express the reversal of w in terms of x^R and y.]

(b) Use structural induction to prove that $(w_1w_2)^R = w_2^R w_1^R$.

Solution : (a) The string of length 0, namely the empty string, is its own reversal, so we define $\lambda^R = \lambda$. A string w of length $n+1$ can always be written as vy , where v is a string of length n (the first n symbols of w), and y is a symbol (the last symbol of w). To reverse w, we need to start with y , and then follow it by the first part of w (namely v), reversed. Thus we define $w^R = y(v^R)$. (Note that the parentheses are for our benefit–they are not part of the string.)

(b) We induct on w_2 . The basis step is $(w_1\lambda)^R = w_1^R = \lambda w_1^R = \lambda^R w_1^R$. For the inductive step, assume that $w_2 = w_3x$, where w_3 is a string of length one less than the length of w_2 , and x is a symbol (the last symbol of w_2). Then we have $(w_1w_2)^R = (w_1w_3x)^R = x(w_1w_3)^R$ (by the recursive definition given in the solution of part(a)). This in turn equals $xw_3^R w_1^R$ by the inductive hypothesis, which is $(w_3x)^R w_1^R$ (again by the definition). Finally, this equals $w_2^R w_1^R$, as desired.

Question 4: Show that a simple graph G with n vertices is connected if it has more than $(n-1)(n-2)/2$ edges.

Solution : Before we give a correct proof here, let us look at an incorrect proof that students often give for this exercise. It goes something like this. "Suppose that the graph is not connected. Then no vertex can be adjacent to every other vertex, only to $n-2$ other vertices. One vertex joined to $n-2$ other vertices creates a component with $n-1$ vertices in it. To get the most edges possible, we must use all the edges in this component. The number of edges in this component is thus $C(n-1,2) = (n-1)(n-2)/2$, and the other component (with only one vertex) has no edges. Thus we have shown that a disconnected graph has at most $(n-1)(n-2)/2$ edges, so every graph with more edges than that has to be connected." The fallacy here is in assuming–without justification–that the maximum number of edges is achieved when one component has $n-1$ vertices. What if, say, there were two components of roughly equal size? Might they not together contain more edges? We will see that the answer is "no", but it is important to realize that this requires proof–it is not obvious without some calculations.

Here is a correct proof, then. Suppose that the graph is not connected. Then it has a component with k vertices in it, for some k between 1 and $n-1$, inclusive. The remaining $n-k$ vertices are in one or more other components. The maximum number of edges this graph could have is then $C(k,2)+C(n-k,2)$, which, after a bit of algebra, simplifies to $k^2-nk+(n^2-n)/2$. This is a quadratic function of k . It is minimized when $k=n/2$ (the k coordinate of the vertex of the parabola that is the graph of this function) and maximized at the endpoints of the domain, namely $k=1$ and $k=n-1$. In the latter cases its value is $(n-1)(n-2)/2$. Therefore the largest number of edges that a disconnected graph can have is $(n-1)(n-2)/2$, so every graph with more edges than this must be connected.