CS1340: DISCRETE STRUCTURES II

PRACTICE QUESTIONS II - SOLUTIONS

(1) *Rosen - 2012 ed, Section 8.2, Q 21* What is the general form of the solutions of a linear homogeneous recurrence relation if its characteristic equation has roots $1, 1, 1, 1, -2, -2, -2, 3, 3, -4$?
Proof: We have four roots with multiple multiplicities – $r_1 = 1$ with 4 as its multiplicity, $r_2 = -2$ with 3 as its multiplicity, $r_3 = 3$ with 2 as its multiplicity, $r_4 = -4$ with 1 as its multiplicity.
The general solution is : $a_n = (\alpha_{1,0} + \alpha_{1,1}n + \alpha_{1,2}n^2 + \alpha_{1,3}n^3)(1)^n + (\alpha_{2,0} + \alpha_{2,1}n + \alpha_{2,2}n^2)(-2)^n + (\alpha_{3,0} + \alpha_{3,1}n)3^n + (\alpha_{4,0})(-4)^n$.

(2) *Rosen - 2012 ed, Section 8.2, Q 27* What is the general form of the particular solution of the linear nonhomogeneous recurrence relation $a_n = 8a_{n-2} - 16a_{n-4} + g(n)$ if $g(n) = n^3$?
Characteristic equation $r^4 - 8r^2 + 16 = 0$. Factor to find roots: $(r^2-4)(r^2-4) = 0$. Our roots are $r_1 = 2$ with multiplicity 2 and $r_2 = -2$ with multiplicity 2. So the particular solution will be of the form $f^{(p)}(n) = (an^3 + bn^2 + cn + d)$ and we put it in the recurrence to solve for a, b, c, d. The answers can be calculated by solving for them. I will give the exact solutions in class.

(3) *Rosen - 2012 ed, Section 8.2, Q 33* Find the solution of the recurrence relation $a_n = 4a_{n-1} - 4a_{n-2} + (n + 1)2^n$.

Answer: Need to start guessing from $(an+b)2^n$, then quadratic with $(an^2 + bn + c)2^n$ and then finally the cubic polynomial times 2^n. The final answer is $a_n = (\alpha_1 + \alpha_2 n + n^2 + \frac{n^3}{6}) \cdot 2^n$.

Note: The question you might ask is will this method always give rise to a consistent set of coefficients. If $g(n)$ is of the form $r^n v(n)$ where r is a fixed real number and $v(n)$ is a polynomial in n with real coefficients then yes you can always find a particular solution. ***But if $g(n)$ is of any other form, it may not work****. For example, if $g(n)$ is of the form log n then the method of undetermined coefficients will not work. Consider, $x_{n+1} - x_n = log\ n$. You cannot find a particular solution using log n, nlog n, n^2log n, etc. This is the case even when the recurrence relation has constant coefficients. The results are more complicated if you consider other coefficients like polynomials.*

You can also use the result from Theorem 6 in Sect 8.2. The proof of that result is slightly involved and for those who of you who are interested in reading it up, you can find the detailed proof of the result in H.F.Mattson Discrete Mathematics with Applications. It is the proof of Theorem 6.5 in Chapter 11.

(4) *Rosen - 2012 ed, Section 8.3, Q 17* Suppose that the votes of n people for different candidates (where there can be more than two candidates) for a

particular office are the elements of a sequence. A person wins the election if this person receives a majority of the votes.

(a) Devise a divide-and-conquer algorithm that determines whether a candidate received a majority and, if so, determine who this candidate is. [Hint: Assume that n is even and split the sequence of votes into two sequences, each with $n/2$ elements. Note that a candidate could not have received a majority of votes without receiving a majority of votes in at least one of the two halves.]

(b) Use the master theorem to give a big-O estimate for the number of comparisons needed by the algorithm you devised in part (a).

Proof: a)If the sequence has only one element then the person is the winner. For the recursive step, divide the list into two parts as equally as possible. As given in the hint it is not possible for a candidate to receive a majority of votes without receiving a majority of votes in at least one of the two halves. Why? Since if a candidate got less than or equal to half votes in each half then he got less than or equal to half votes in all. Apply the algorithm recursively to each half to come up with atmost two names. Then run through the entire list to count the number of occurences of each of those names to decide which if either is the winner. This requires at most $2n$ comparisons for a list of length n. $T(n) = 2T(n/2) + 2n$.

b)$a = 2, b = 2, c = 2, d = 1$. We have $a = b^d$ we have number of comparisons as $O(nlogn)$.

(5) *Rosen - 2012 ed, Section 8.3, Q 22* Suppose that the function f satisfies the recurrence relation $f(n) = 2f(\sqrt{n}) + logn$ where n is a perfect square greater than 1 and $f(2) = 1$. Find a big-0 estimate for $f(n)$. *Make the substitution* $m = logn$.

Make the substitution $m = logn$. Rewriting the recurrence relation, $f(2^m) = 2f(2^{m/2}) + m$ with $f(2) = 1$. Substitute $T(m) = f(2^m)$, we have $T(m) = 2T(m/2) + m$ with $T(1) = 1$. We have $a = 2, b = 2, d = 1$ and $a = b^d$ so by Master Theorem, $T(m) = O(mlogm)$.

$T(m) = f(2^m) = f(2^{logn}) = f(n)$ and $m = logn$: $f(n) is O(lognloglogn)$.