MTH 352 10-18-17

§7.2 Solve

#171 $a_n = 2a_{n-1} + 8a_{n-2}$ $a_0 = 4,\ a_1 = 10$

Sol: $t^2 - 2t - 8 = 0$ $(t^2 = 2t + 8)$

$(t-4)(t+2) = 0$

$r_1 = 4,\ r_2 = -2$

$a_n = P(4)^n + Q(-2)^n$

$+2(P + Q = 4) \Rightarrow +2P + 2Q = +8$

$4P - 2Q = 10$ $\qquad 4P - 2Q = 10$

$6P = 18 \Rightarrow \boxed{P = 3}$

$\boxed{Q = 1}$

$a_n = 3(4)^n + (-2)^n$

$a_0 = 3(4)^0 + (-2)^0 = 3 + 1 = 4$ ✓

$a_1 = 3(4) + (-2) = 10$ ✓

MTH 356 10-18-17

§7.2 Solve

#17) $a_n = 2a_{n-1} + 8a_{n-2}$ $a_0 = 4,\ a_1 = 10$

Sol: $t^2 - 2t - 8 = 0$ $(t^2 = 2t + 8)$

$(t-4)(t+2) = 0$

$r_1 = 4,\ r_2 = -2$

$a_n = P(4)^n + Q(-2)^n$

$+2(P + Q = 4) \Rightarrow +2P + 2Q = +8$

$4P - 2Q = 10$ $\quad 4P - \cancel{2Q} = 10$

$6P = 18 \Rightarrow \boxed{P = 3}$

$\boxed{Q = 1}$

$a_n = 3(4)^n + (-2)^n$

$a_0 = 3(4)^0 + (-2)^0 = 3 + 1 = 4$ ✓

$a_1 = 3(4) + (-2) = 10$ ✓

Show $a_n = 2a_{n-1} + 8a_{n-2}$ and

$a_n = 3(4)^n + (-2)^n$ produce the same sequence of values.

Pf (induction)

BC. ✓

IC: Assume $a_r = 3(4)^r + (-2)^r$ is true for all r $1 \le r \le k$, for some $k \geq 0$. We need to show that

$$a_{k+1} = 3(4)^{k+1} + (-2)^{k+1}.$$

Note that $a_{k+1} = 2a_k + 8a_{k-1}$ by RR

$$= 2\left(3(4)^k + (-2)^k\right) + 8\left(3(4)^{k-1} + (-2)^{k-1}\right)$$

$$= 3\left(\frac{2}{4}\right)4^{k+1} + (-1)(-2)^{k+1} + 3\left(\frac{8}{16}\right)(4)^{k+1} + \frac{8}{4}(-2)^{k+1}$$

$$= 3(4)^{k+1}\underbrace{\left(\frac{1}{2} + \frac{1}{2}\right)}_{=1} + (-2)^{k+1}\underbrace{(-1+2)}_{=1}$$

$$a_{k+1} = 3(4)^{k+1} + (-2)^{k+1}.$$

Therefore, by the PMI, the two equations produce the same sequence of numbers. ∎

worst-case analysis of binary search algo

Define the worst-case required by the binary search = The number of times that the algo is invoked in the worst case for a sequence containing n items, denoted by a_n.

Initial condition: $n=1$, then the sequence has only 1 element namely s_i with $i=j$.

In the worst-case $key \neq s_k$ and hence the algo is called a second time at line 11. This time we have $i=2$ at line 10, and hence $i>j$ at line 2, and the algo terminates at line 3.

Hence if $n=1$, the algo is called twice, i.e., $a_1=2$.

Suppose $n>1$ is an arbitrary number.

At line 2, $i>j$ is false. At line 5, key is not found. At line 11, the algo is called, by definition this call will require a total of a_m invocations where m is the size of the input sequence at line 11. Note that the size of the left and right subsequence are $\lfloor \frac{n-1}{2} \rfloor$ and $\lfloor \frac{n}{2} \rfloor$ respectively.

Then, for the worst case, we choose the larger subsequence as the input. Hence the total number of invocations at line 11 will be $a_{\lfloor \frac{n}{2} \rfloor}$.

Thus the total number of invocations is

$$(1) \quad a_n = 1 + a_{\lfloor \frac{n}{2} \rfloor} \ldots$$

Goal: Find Θ notation for the given recurrence relation. $a_n = 1 + a_{\lfloor \frac{n}{2} \rfloor}$ (1) $a_1 = 2$

Case 1: suppose n is a power of 2, i.e., $n = 2^k$ for some positive integer k. $\frac{n}{2} = \frac{2^k}{2} = 2^{k-1}$

Then (1) becomes $a_{2^k} = 1 + a_{2^{k-1}}$, $k = 1, 2, \ldots$

use backtracking to solve RR.

$a_{2^{k-1}} = a_{2^{k-2}} + 1$

$a_{2^{k-2}} = a_{2^{k-3}} + 1$

$$\begin{aligned}
a_{2^k} &= a_{2^{k-1}} + 1 \\
&= (a_{2^{k-2}} + 1) + 1 \\
&= a_{2^{k-2}} + 2 \\
&= (a_{2^{k-3}} + 1) + 2 = a_{2^{k-3}} + 3 \\
&\;\;\vdots \\
&= a_{2^{k-k}} + k \\
&= a_{2^0} + k
\end{aligned}$$

$$a_{2^k} = a_1 + k$$

$$a_{2^k} = 2 + k$$

Since $n = 2^k$, we have $k = \lg n$

Therefore $\boxed{a_n = 2 + \lg n}$.