CSCE 222 (Carlisle), Discrete Structures for Computing
Spring 2022
Homework 10

Type your name below the pledge to sign
On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work.
HUY QUANG LAI

Instructions:

- The exercises are from the textbook. You are encouraged to work extra problems to aid in your learning; remember, the solutions to the odd-numbered problems are in the back of the book.
- Grading will be based on correctness, clarity, and whether your solution is of the appropriate length.
- Always justify your answers.
- Don't forget to acknowledge all sources of assistance in the section below, and write up your solutions on your own.
- *Turn in .pdf file to Gradescope by the start of class on Monday, April 4, 2022.* It is simpler to put each problem on its own page using the LaTeX clearpage command.

Help Received:

- Rosen, Kenneth H. *Discrete Mathematics and Its Applications*. McGraw-Hill, 2019.

Exercises for Section 8.1:

8(a-c): (2 points).

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

Consider a string of length $n \geq 3$ that contains three consecutive 0s. Such a string either ends with 1, or with 10, or with 100, or with 000. In the first case, there are s_{n-1} possibilities. In the second case, there are s_{n-2} possibilities. In the third case, there are s_{n-3} possibilities. And, in the fourth case, there are 2^{n-3} possibilities. Hence the recurrence relation is

$$s_n = s_{n-1} + s_{n-2} + s_{n-3} + 2^{n-3}, n \geq 3$$

What are the initial conditions?
$s_0 = s_1 = s_2 = 0$

How many bit strings of length seven contain three consecutive 0s?
$s_3 = s_2 + s_1 + s_0 + 2^0 = 1$
$s_4 = s_3 + s_2 + s_1 + 2^1 = 3$
$s_5 = s_4 + s_3 + s_1 + 2^2 = 8$
$s_6 = s_5 + s_4 + s_3 + 2^3 = 20$
$s_7 = s_6 + s_5 + s_4 + 2^4 = 47$

20(a-b): (2 points).

A bus driver pays all tolls, using only nickels and dimes, by throwing one coin at a time into the mechanical toll collector.

Find a recurrence relation for the number of different ways the bus driver can pay a toll of n cents (where the order in which the coins are used matters).

Let P_n be the number of ways the bus driver can pay a toll of n cents by using nickels and dimes (where the order in which the coins are used matters).
If the toll can be divided by 5, i.e. $n = 5k$ where $k \in \mathbb{N}$, then the number of ways the bus driver can pay the toll is P_{5k}.

If the toll cannot be divided by 5, i.e. $n = 5k + 1, 5k + 2, 5k + 3$, or $5k + 4$ where $k \in \mathbb{N}$, then the number of ways the bus driver can pay the toll is $P_{5(k+1)}$

Let $P_m = P_{5k}$
Then $P_{5(k-1)} = P_{m-1}$ and $P_{5(k-2)} = P_{m-2}$
Therefore, the recurrence relation is $P_m = P_{m-1} + P_{m-2}$, where $P_0 = 1, P_1 = 1$

In how many different ways can the driver pay a toll of 45 cents?

$m = 9$
$P_0 = 1$
$P_1 = 1$
$P_2 = P_0 + P_1 = 2$
$P_3 = P_1 + P_2 = 3$
$P_4 = P_2 + P_3 = 5$
$P_5 = P_3 + P_4 = 8$
$P_6 = P_4 + P_5 = 13$
$P_7 = P_5 + P_6 = 21$
$P_8 = P_6 + P_7 = 34$
$P_9 = P_7 + P_8 = 55$

56(b-e): (3 points).

In this exercise we will develop a dynamic programming algorithm for finding the maximum sum of consecutive terms of a sequence of real numbers. That is, given a sequence of real numbers $a_1, a_2, \cdots, a_n$, the algorithm computes the maximum sum $\sum_{i=j}^{k} a_i$ where $1 \leq j \leq k \leq n$.

Let $M(k)$ be the maximum of the sums of consecutive terms of the sequence ending at a_k. That is, $M(k) = \max_{1\leq j\leq k} \sum_{i=j}^{k} a_i$. Explain why the recurrence relation $M(k) = \max(M(k-1) + a_k, a_k)$ holds for $k = 2, \cdots, n$.

$M(k) = \max(M(k-1) + a_k, a_k)$
$M(k) = \max(\max(M(k-2 + a_{k-1}, a + k - 1)) + a_k, a_k)$ $k = 1$ is not allowed since $k - 1 = 0$ and there is no such term for $M(0)$. Looking at the recursion, we find that there are all sums $a_k, a_k + a_{k-1}, \cdots a_k + \cdot + a_1$ in the set inside max function. Hence, the iteration will give the maximum of sums consecutive terms of the sequence ending at a_k, and the recurrence relation holds.

Use part (b) to develop a dynamic programming algorithm for solving this problem.

```
m=a_1
for i:=2 to n:
    t=a_i
    for j:=i to 2:
        t = t+a_{j+1}
        m = max(m,t,a_j)
return m
```

Show each step your algorithm from part (c) uses to find the maximum sum of consecutive terms of the sequence $2, -3, 4, 1, -2, 3$.

$a_1 = 2, a_2 = -3, a_3 = 4, a_4 = 1, a_5 = -2, a_6 = 3$

Starting with algorithm considering the first and second term:
$m = 2, i = 2, t = -3$
Running j-loop one time
$j = 2, t = 2 + (-3) = -1$
$m = \max(2, -3, 2) = 2$

Running algorithm with first three terms:
$i = 3, t = 4$
Running j-loop two times
$j = 3, t = 4 + (-3) = 1$
$m = \max(2, 1, -3)$

$j = 2, t = 1 + 2 = 3$
$m = \max(2, 3, -3) = 3$

Running algorithm with first four terms:
$i = 4, t = 1$
Running j-loop three times
$j = 4, t = 1 + 4 = 5$
$m = \max(3, 5, 1) = 5$

$j = 3, t = 5 + (-3) = 2$
$m = \max(5, 2, 1) = 5$

$j = 2, t = 2 + 2 = 4$
$m = \max(5, 4, 1) = 5$

Running algorithm with first five terms:
$i = 5, t = -2$
Running j-loop four times
$j = 5, t = -1 + 4 = 3$
$m = \max(5, -1, 2) = 5$

$j = 4, t = -1 + 4 = 3$
$m = \max(5, 3, -2) = 5$

$j = 3, t = 3 + (-3) = 0$
$m = \max(5, 0, -2) = 5$

$j = 2, t = 0 + 2 = 2$
$m = \max(5, 2, -2) = 5$

Running algorithm with first six terms:
$i = 6, t = 3$ Running j-loop five times
$j = 6, t = 3 + (-2) = 1$
$m = \max(5, 1, 3) = 5$

$j = 5, t = 1 + 1 = 2$
$m = \max(5, 2, -2) = 5$

$j = 4, t = 2 + 4 = 6$

$m = \max(5, 6, -2) = 6$

$j = 3, t = 6 + (-3) = 3$
$m = \max(6, 3, -2) = 6$

$j = 3, t = 3 + 2 = 5$
$m = \max(6, 5, -2) = 6$

The maximum sum of consecutive terms is 6.

Show that the worst-case complexity in terms of the number of additions and comparisons of your algorithm from part (c) is linear.

Starting with $n = 2$ there is one addition and one comparison.
Next, for $n = 3$, there are two additions and two comparisons.
Next, for $n = 4$, there are three additions and three comparisons.
Next, for $n = 5$, there are four additions and four comparisons.

As n increases by one, the number of additions and comparisons also increases by 1. Because of this relationship, the worse-case complexity of the algorithm is linear.

Exercises for Section 8.2:

4(a,c,e,g): (2 points).

Solve these recurrence relations together with the initial conditions given.

$a_n = a_{n-1} + 6a_{n-2}$ for $n \geq 2, a_0 = 3, a_1 = 6$
$a_n - a_{n-1} - 6_{n-2} = 0$
$r^2 - r - 6r = 0$
$(r-3)(r+2)$
$r = \{-2, 3\}$
$a_n = c(-2)^n + d(3)^n \; c + d = 3, -2c + 3b = 6$
$c = \frac{3}{5}, d = \frac{12}{5}$
$a_n = \frac{3}{5}(-2)^n + \frac{12}{5}(3)^n$

$a_n = 6a_{n-1} - 8a_{n-2}$ for $n \geq 2, a_0 = 4, a_1 = 10$
$a_n - 6a_{n-1} + 8a_{n-2} = 0$
$r^2 - 6r + 8 = 0$
$(r-4)(r-2) = 0$
$r = \{2, 4\}$
$a_n = c(2)^n + d(4)^n$
$c + d = 4, 2c + 4d = 10$
$c = 3, d = 1$
$a_n = 3(2)^n + 1(4)^n$

$a_n = a_{n-2}$ for $n \geq 2, a_0 = 5, a_1 = -1$
$a_n - a_{n-2} = 0$
$r^2 - 1 = 0$
$r = \pm 1$
$a_n = c(-1)^n + d(1)^n$
$c + d = 5, -c + d = -1$
$c = -3, d = 7$
$a_n = -3(-1)^n + 7(1)^n$

$a_{n+2} = -4a_{n+1} + 5a_n$ for $n \geq 0, a_0 = 2, a_1 = 8$
$a_{n+2} + 4a_{n+1} - 5a_n = 0$
$r + 4r^2 - 5 = 0$
$(r-1)(r+5) = 0$
$r = \{-5, 1\}$
$a_n = c(-5)^n + d(1)^n$
$c + d = 2, -5c + d = 10$
$c = -1, d = 3$
$a_n = -1(-5)^n + 3(1)^n$

28(a-b): (2 points).

Find all solutions of the recurrence relation $a_n = 2a_{n-1} + 2n^2$.

$a_n - 2a_{n-1} = 0$
$r - 2 = 0, r = 2$
$a_n = c(2)^n$

$a_n = c(2)^n + p_2 n^2 + p_1 n + p_0$

$$\begin{aligned} a_n &= 2a_{n-1} + 2n^2 \\ p_2 n^2 + p_1 n + p_0 &= 2(p_2(n-1)^2 + p_1(n-1) + p_0) + 2n^2 \\ (p_2 + 2)n^2 + (p_1 - 4p_2)n + (2p_2 - 2p_1 + p_0) &= 0 \\ (p_2 + 2)n^2 + (p_1 - 4p_2)n + (2p_2 - 2p_1 + p_0) &= 0n^2 + 0n + 0 \end{aligned}$$

$p_2 + 2 = 0, p_2 = -2$
$p_1 - 4p_2 = 0, p_1 = -8$
$2p_2 - 2p_1 + p_0 = 0$
$-4 + 16 + p_0, p_0 = -12$

$a_n = c(2)^n - 2n^2 - 8n - 12$

Find the solution of the recurrence relation in part (a) with initial condition $a_1 = 4$.
$4 = c(2)^1 - 2(1^2) - 8 - 12$
$4 = 2c - 22$
$c = 13$

$a_n = 13 \cdot 2^n - 2n^2 - 8n - 12$

Exercises for Section 8.3:

18(a-b): (2 points).

Suppose that each person in a group of n people votes for exactly two people from a slate of candidates to fill two positions on a committee. The top two finishers both win positions as long as each receives more than $\frac{n}{2}$ votes.

Devise a divide-and-conquer algorithm that determines whether the two candidates who received the most votes each received at least $\frac{n}{2}$ votes and, if so, determine who these two candidates are.

The base case is that a sequence with one element means that the one person on the list is the winner.
For the recursive step, divide the list into two equal parts and count which name occurs the most in the two parts.
The winner requires a majority of votes and will need at least $\frac{n}{2} + 1$ votes.
Keep applying this recursive step to each half until the list contains at most two names.
Count the number of occurrences in the whole list of the two remaining names and this will decide the winner.

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

The function would be $f(n) = 2f\left(\frac{n}{2}\right) + 2n$. So $a = 2, b = 2, c = 2, d = 1$.
By the master theorem, $a = b^d$ so the big-O is $O(n^d \log n) = O(n \log n)$

22: (2 points).

Suppose that the function f satisfies the recurrence relation $f(n) = 2f(\sqrt{n}) + \log n$ whenever n is a perfect square greater than 1 and $f(2) = 1$.

Find $f(16)$.
$f(4) = 2f(2) + \log 4 = 4$
$f(16) = 2f(4) + \log 16$
$f(16) = 12$

Find a big-O estimate for $f(n)$.
Let $m = \log n, n = 2^m$
$f(2^m) = 2f(\sqrt{2^m}) + \log{(2^m)}$
$f(2^m) = 2f\left(2^{\frac{m}{2}}\right) + m$

$T(m) = 2T\left(\frac{m}{2}\right) + m$
$a = 2, b = 2, d = 1$
$a = b^d$
$O(m^d \log m)$
$O(\log n \cdot \log{(\log n)})$

Exercises for Section 8.4:

12(a,c,e): (3 points).

Find the coefficient of x^{12} in the power series of each of these functions.

$$\frac{1}{1+3x}$$

$$\begin{aligned}\frac{1}{1+3x} &= \frac{1}{1-(-3x)} \\ &= \sum_{k=0}^{+\infty}(-3x)^k\end{aligned}$$

$(-3)^{12}$

$$\frac{1}{(1+x)^8}$$

$$\begin{aligned}\frac{1}{(1+x)^8} &= \frac{1}{(1-(-x))^8} \\ &= \sum_{k=0}^{+\infty}\binom{7+k}{k}(-x)^k\end{aligned}$$

$\binom{19}{12}(-1)^{12}$

$$\frac{x^3}{(1+4x)^2}$$

$$\begin{aligned}\frac{x^3}{(1+4x)^2} &= x^3 \cdot \frac{1}{(1-(-4x))^2} \\ &= x^3 \cdot \sum_{k=0}^{\infty}\binom{1+k}{k}(-4x)^k \\ &= \sum_{k=0}^{\infty}\binom{1+k}{k}(-4)^k x^{k+3}\end{aligned}$$

$\binom{10}{9}(-4)^9$

14: (2 points).

Use generating functions to determine the number of different ways 12 identical action figures can be given to five children so that each child receives at most three action figures.

$(1 + x + x^2 + x^3)^5$ coefficient of x^{12}
There are 35 ways of distributing 12 identical action figures to five children so that each child receives at most three action figures