Solution Assignment 4

6.

Find the number of primes ≤ 100.

SOLUTION:
Let $S = \{n \in \mathbb{N} \mid 1 < n \leq 100\}$. By Theorem 4.2, a positive integer n is a prime if and only if it has no prime factors $\leq \lfloor\sqrt{n}\rfloor$. Therefore, an element in S is prime if and only if it has no prime factors ≤ 10. There are four primes ≤ 10, namely, 2, 3, 5, and 7. Thus the primes ≤ 100 are these four primes, and those integers in S not divisible by 2, 3, 5, or 7.

Let P_2 be the property that an integer in S is divisible by 2, P_3 the property that an integer in S is divisible by 3, P_5 the property that an integer in S is divisible by 5, and P_7 the property that an integer in S is divisible by 7. Then $N(P_2' P_3' P_5' P_7')$ denotes the number of integers in S not divisible by 2, 3, 5, or 7. Thus there are $4 + N(P_2' P_3' P_5' P_7')$ primes in S.

To find $N(P_2' P_3' P_5' P_7')$: First notice that $|S| = 99$. Secondly, let $r, s, t \in \{2, 3, 5, 7\}$. Since r and s are primes, an integer has property $P_r P_s$ if it has both properties P_r and P_s. This process can be extended to $P_r P_s P_t$ and $P_2 P_3 P_5 P_7$. Therefore, the number of elements in S having property $P_r P_s$ is given by $\lfloor 100/rs \rfloor$, the number of elements in S having property $P_r P_s P_t$ is given by $\lfloor 100/rst \rfloor$, and those with $P_2 P_3 P_5 P_7$ by $\lfloor 100/2 \cdot 3 \cdot 5 \cdot 7 \rfloor$.

By the alternate inclusion–exclusion principle,

$$
\begin{aligned}
N(P_1', P_2', \ldots, P_n') &= |S| - \sum N(P_i) + \sum N(P_i P_j) - \sum N(P_i P_j P_k) \\
&\quad + N(P_2 P_3 P_5 P_7) \\
&= 99 - [N(P_2) + N(P_3) + N(P_5) + N(P_7)] \\
&\quad + [N(P_2P_3) + N(P_2P_5) + N(P_2P_7) + N(P_3P_5) + N(P_3P_7) \\
&\quad + N(P_5P_7)] - [N(P_2P_3P_5) + N(P_2P_3P_7) + N(P_3P_5P_7)] \\
&\quad + N(P_2P_3P_5P_7) \\
&= 99 - (\lfloor 100/2 \rfloor + \lfloor 100/3 \rfloor + \lfloor 100/5 \rfloor + \lfloor 100/7 \rfloor) + (\lfloor 100/2\cdot 3 \rfloor \\
&\quad + \lfloor 100/2\cdot 5 \rfloor + \lfloor 100/2\cdot 7 \rfloor + \lfloor 100/3\cdot 5 \rfloor + \lfloor 100/3\cdot 7 \rfloor \\
&\quad + \lfloor 100/5\cdot 7 \rfloor) - (\lfloor 100/2\cdot 3\cdot 5 \rfloor + \lfloor 100/2\cdot 3\cdot 7 \rfloor + \lfloor 100/3\cdot 5\cdot 7 \rfloor) \\
&\quad + \lfloor 100/2\cdot 3\cdot 5\cdot 7 \rfloor \\
&= 99 - (50 + 33 + 20 + 14) + (16 + 10 + 7 + 6 + 4 + 2) \\
&\quad - (3 + 2 + 1) + 0 \\
&= 21
\end{aligned}
$$

Thus, there are $4 + 21 = 25$ primes ≤ 100. ∎

7.

Find the total number of submatrices of an $m \times n$ matrix.

SOLUTION:
Any r rows can be selected in $\binom{m}{r}$ ways. So, by Theorem 6.17, the total number of combinations of rows from m rows equals $\sum_{r=1}^{m} \binom{m}{r} = 2^m - 1$, Similarly, the total number of columns we can choose is $2^n - 1$. Thus there are $(2^m - 1)(2^n - 1)$ ways of choosing rows and columns; that is, there are $(2^m - 1)(2^n - 1)$ submatrices in an $m \times n$ matrix. ∎

13.

A salesperson at a computer store would like to display six models of personal computers, five models of computer monitors, and four models of keyboards. In how many different ways can he arrange them in a row if items of the same family are to be next to each other?

SOLUTION:
There are three types of items: personal computers, monitors, and keyboards. Think of the items in each family as *tied together* into one unit. These families can be arranged in $P(3, 3) = 3!$ ways. Now the items within each family can be rearranged. The six models of personal computers can be arranged in $P(6, 6) = 6!$ ways, the monitors in $P(5, 5) = 5!$ ways, and the keyboards in $P(4, 4) = 4!$ different ways. Thus, by the multiplication principle, the total number of possible arrangements is $3!\,6!\,5!\,4! = 12{,}441{,}600$. ■

17.

Write a recursive algorithm to compute the nth Fibonacci number F_n.

SOLUTION:
Recall from Example 5.7 that the recursive definition of F_n involves two initial conditions $F_1 = 1 = F_2$, and the recurrence relation $F_n = F_{n-1} + F_{n-2}$, where $n \geq 3$. These two cases can be combined into straightforward Algorithm 5.4.

```
Algorithm Fibonacci(n)
(* This algorithm computes the nth Fibonacci number
   using recursion. *)
 Begin (* algorithm *)
   if n = 1 or n = 2 then     (* base cases *)
     Fibonacci ← 1
   else                       (* general case *)
     Fibonacci ← Fibonacci(n - 1) + Fibonacci(n - 2)
 End (* algorithm *)
```

18.

Write a recursive algorithm to compute the gcd of two positive integers x and y.

SOLUTION:
If $x > y$, $\gcd\{x,y\} = \gcd\{x - y, y\}$. (See Exercise 34 in Section 4.2.) We use this fact to write Algorithm 5.5.

```
Algorithm gcd(x,y)
(* This algorithm computes the gcd of two positive
   integers x and y using recursion. *)
 Begin (* algorithm *)
   if x > y then
      gcd ← gcd{x - y, y}
   else if x < y then
           gcd ← gcd{y, x}
        else
           gcd ← x
 End (* algorithm *)
```

19.

(Binary Search Algorithm) Write a recursive algorithm to search an ordered list X of n items and determine if a certain item (*key*) occurs in the list. Return the location of *key* if the search is successful.

SOLUTION:
Because the algorithm is extremely useful, we first outline it:

```
compute the middle index.
if key = middle value then
  we are done and exit
else if key < middle value then
        search the lower half
     else
        search the upper half.
```

The algorithm is given in Algorithm 5.6.

```
Algorithm binary search(X,low,high,key,found,mid)
(* The algorithm returns the location of key in the
   variable mid in the list X if the search is successful.
```

```
 Low, mid, and high denote the lowest, middle, and highest
 indices of the list. Found is a boolean variable;
 it is true if key is found and false otherwise. *)
Begin (* algorithm *)
  if low ≤ high then (* list is nonempty *)
    begin (* if *)
      found ← false (* boolean flag *)
      mid ← ⌊(low + high)/2⌋
      if key = x_mid then
        found ← true (* we are done. *)
      else
        if key < x_mid then (* search the lower half *)
          binary search(X,low,mid - 1,key,found,mid)
        else (* search the upper half *)
          binary search(X,mid + 1,high,key,found,mid)
  endif
End (* algorithm *)
```

20.

SOLUTION:
Recall that solving the puzzle involves three steps:

- Move the top $n - 1$ disks from X to Y using Z as an auxiliary peg;
- Move disk n from X to Z; and
- Move the $n - 1$ disks from Y to Z using X as an auxiliary.

We also must count the moves made. The resulting Algorithm 5.3 follows.

```
Algorithm tower (X,Z,Y,n,count)
(* This algorithm, using recursion, prints the various moves
   needed to solve the Tower of Brahma puzzle and returns
   the total number of moves needed in the global variable count.
   Count must be initialized to 0 in the calling module. *)
Begin (* algorithm *)
  if n = 1 then (* base case *)
    begin (* if *)
      move disk 1 from X to Z
      count ← count + 1
    endif
  else (* general case *)
    begin (* else *)
      tower(X,Y,Z,n - 1,count) (* move the top n - 1 disks *)
      move disk n from X to Z
      count ← count + 1
      tower(Y,Z,X,n - 1,count)
    endelse
End (* algorithm *)
```

23.

Let $a, b \in \mathbb{R}$ and $b \neq 0$. Let α be a real or complex solution of the equation $x^2 - ax - b = 0$ with degree of multiplicity two. Then $a_n = A\alpha^n + Bn\alpha^n$ is the general solution of the LHRRWCC $a_n = aa_{n-1} + ba_{n-2}$.

PROOF:
Since α is a root of the equation $x^2 - ax - b = 0$ with degree of multiplicity two,

$$\begin{aligned} x^2 - ax - b &= (x - \alpha)^2 \\ &= x^2 - 2\alpha x + \alpha^2 \end{aligned}$$

Therefore,

$$a = 2\alpha \quad \text{and} \quad b = -\alpha^2 \tag{5.12}$$

- *To show that $a_n = n\alpha^n$ satisfies the recurrence relation*:
  Notice that

$$\begin{aligned} aa_{n-1} + ba_{n-2} &= a[(n-1)\alpha^{n-1}] + b[(n-2)\alpha^{n-2}] \\ &= 2\alpha[(n-1)\alpha^{n-1}] + (-\alpha^2)[(n-2)\alpha^{n-2}] \\ & \qquad \text{by (5.12)} \\ &= \alpha^n[2(n-1) - (n-2)] \\ &= n\alpha^n = a_n \end{aligned}$$

  Therefore, $n\alpha^n$ is a solution of the recurrence relation.

Then $a_n = A\alpha^n + Bn\beta^n$ is the general solution of the given recurrence relation, where A and B are selected in such a way that the initial conditions are satisfied. (The values of A and B can be found using initial conditions, as in Theorem 5.2.) ■

25. (a)

Let $\gcd\{a,b\} = d$. When $n = 0$, $\gcd\{x_n,y_n\} = \gcd\{x_0,y_0\} = \gcd\{x,y\}$. $\therefore P(0)$ is true. Assume $P(k)$ is true: $\gcd\{x_k,y_k\} = \gcd\{x,y\} = d$. Then $x_k = y_{k-1}, y_k = r_{k-1}, r_k = x_k \bmod y_k$, and $x_k = q_k y_k + r_k$. (1)
Since $d|x_k$ and $d|y_k$, $d|r_k$.
To show that P(k + 1) is true: $\gcd\{x_{k+1}, y_{k+1}\} = d$.
Let $\gcd\{x_{k+1}, y_{k+1}\} = d'$. (2)
$x_{k+1} = y_k$, $y_{k+1} = r_k$. $\therefore d|x_{k+1}$ and $d|y_{k+1}$. So $d|d'$. (3)
From (1), $d'|x_{k+1}$ and $d'|y_{k+1}$. $\therefore d'|y_k$ and $d'|r_k$. $\therefore d'|x_k$ by (1). Thus $d'|x_k$ and $d'|y_k$. $\therefore d'|d$. (4)
Thus by (3) and (4), $d = d'$. Thus $P(n)$ is a loop invariant.

26.

PROOF (by strong induction):
Let P(n): The algorithm works for every ordered list of size n.

Basis step When $n = 0$, $low = 1$ and $high = 0$. Since $low \leq high$ is false in line 3, the **while** loop is not executed. So the algorithm returns the correct value 0 from line 14, as expected, and P(0) is true.

Induction step Assume P(i) holds for every $i \leq k$, where $k \geq 0$; that is, the algorithm returns the correct value for any list of size $i \leq k$.

To show that P(k + 1) is true, consider an ordered list X of size k + 1. Since $high = k + 1 \geq 1 = low$, the loop is entered and the middle index is computed in line 5.

Case 1 If $key = x_{\text{mid}}$, we exit the loop (line 7) and the value of *mid* is returned, so the algorithm works.

Case 2 If $key < x_{\text{mid}}$, search the sublist $x_1, \ldots, x_{\text{mid}-1}$; otherwise, search the sublist $x_{\text{mid}+1}, \ldots, x_n$. In both cases, the sublists contain fewer than $k + 1$ elements, so the algorithm works in either case by the inductive hypothesis.

Thus P(k + 1) is true. So, by PMI, P(n) is true for $n \geq 0$; that is, the algorithm works correctly for every ordered list of zero or more items. ■

27.

PROOF (by PMI):
Let $P(n)$: $\text{fact}(n) = n!$, $n \geq 0$.

Basis step When $n = 0$, $\text{fact}(0) = 1 = 1!$ by line 1; so $P(0)$ is true.

Induction step Assume $P(k)$ is true: $\text{fact}(k) = k!$. Then:

$$\begin{aligned}\text{fact}(k+1) &= \text{fact}(k) \cdot (k+1), \text{ by line 6}\\ &= k! \cdot (k+1), \text{ by the inductive hypothesis}\\ &= (k+1)!\end{aligned}$$

Therefore, $P(k+1)$ is true.

Thus, by induction, $P(n)$ holds true for every $n \geq 0$; that is, $P(n)$ is a loop invariant and hence the algorithm correctly computes the value of $n!$, for every $n \geq 0$. ■

28.

Find the number of trailing zeros in 123!

SOLUTION:
By the fundamental theorem of arithmetic, 123! can be factored as 2^a5^bc, where c denotes the product of primes other than 2 and 5. Clearly $a > b$. Each trailing zero in 123! corresponds to a factor of 10 and vice versa.

$$\begin{aligned}\therefore \text{ Number of trailing zeros} &= \begin{pmatrix}\text{Number of products of the form}\\ 2 \cdot 5 \text{ in the prime factorization}\end{pmatrix}\\ &= \text{minimum of } a \text{ and } b\\ &= b, \text{ since } a > b\end{aligned}$$

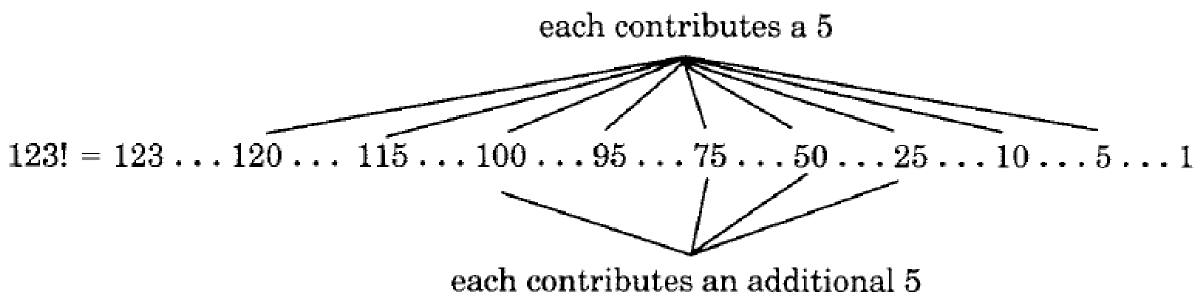
We proceed to find b:

Number of positive integers ≤ 123 and divisible by $5 = \lfloor 123/5 \rfloor = 24$

Each of them contributes a 5 to the prime factorization of 123!

Number of positive integers ≤ 123 and divisible by $25 = \lfloor 123/25 \rfloor = 4$

(See Figure 4.29.) Each of them contributes an additional 5 to the prime factorization. Since no higher power of 5 contributes a 5 in the prime factorization of 123!, the total number of 5's in the prime factorization equals $24 + 4 = 28$. Thus the total number of trailing zeros in 123! is 28.



29.

PROOF:
Consider the following sums:

$$\begin{aligned}
S_1 &= a_1 + a_2 + \cdots + a_k \\
S_2 &= a_2 + a_3 + \cdots + a_{k+1} \\
&\vdots \\
S_n &= a_n + a_1 + \cdots + a_{k-1}
\end{aligned}$$

Each of the first n positive integers appears k times in this set of sums. Then

$$\sum_{i=1}^{n} S_i = k\left(\sum_{i=1}^{n} a_i\right) = k\left(\sum_{i=1}^{n} i\right) = \frac{kn(n+1)}{2},$$

Consider $kn(n+1)/2$ pigeons. We would like to distribute them among n pigeonholes, called $S_1, S_2, \ldots, S_n$. By the generalized pigeonhole principle, at least one of the pigeonholes S_i must contain more than $\lfloor kn(n+1)/2n - 1/n \rfloor = \lfloor [kn(n+1-2)]/2n \rfloor$ pigeons. In other words, $s_i > \lfloor kn(n+1) - 2/2n \rfloor$, as desired. ■

32.

PROOF (by cases):
Consider the n sums $S_i = a_1 + a_2 + \cdots + a_i$, where $1 \leq i \leq n$.

Case 1 If any of the sums S_i is divisible by n, then the statement is true.

Case 2 Suppose none of the sums S_i is divisible by n. When S_i is divided by n, the remainder must be nonzero. So, by the division algorithm, the possible remainders are $1, 2, \ldots, (n-1)$. Since there are n sums and $n-1$ possible remainders, by the pigeonhole principle, two of the sums S_k and S_ℓ must yield the same remainder r when divided by n, where $k < \ell$.

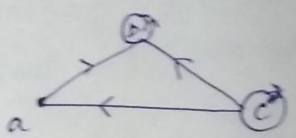
Therefore, there must exist integers q_1 and q_2 such that $a_1 + a_2 + \cdots + a_k = nq_1 + r$ and $a_1 + a_2 + \cdots + a_\ell = nq_2 + r$, where $k < \ell$. Subtracting, we get $a_{k+1} + a_{k+2} + \cdots + a_\ell = n(q_1 - q_2)$. Thus $a_{k+1} + a_{k+2} + \cdots + a_\ell$ is divisible by n. ■

Assignment-4

Ans ① $R = \{(a,b), (b,b), (c,a)$ and $(c,c)\}$ on $\{a,b,c\}$

$R^2 = R \circ R = \{(a,b), (b,b), (c,a), (c,b), (c,c)\}$

$R^3 = R^2 \circ R = \{(a,b), (b,b), (c,a)\ (c,b), (c,c)\} = R^2$



---

② Let $A = \{a, a_2 -- a_m\}$

$B = \{b, b_2 ---- b_n\}$

$C = \{c, c_2 ----- c_p\}$

Then the matrices M_R, M_S, $M_{R\circ S}$ and $M_R \odot M_S$ are of sizes $m \times n$, $n \times P$, $m \times P$, $m \times P$ respectively.

Let $M_{R\circ S} = (x_{ij})$ and $M_R \odot M_S = (y_{ij})$. Then $x_{ij} = 1$ iff $a_i (R \circ S) c_j$. But $a_i (R \circ S) c_j$ iff $a_i R b_k$ and $b_k S c_j$ for some b_k in B.

Thus, $x_{ij} = 1$ iff $y_{ij} = 1$, so $x_{ij} = y_{ij}$ for every i and j.

Consequently $M_{R\circ S} = M_R \odot M_S$.

Q3 → antisymmetric.

Qu5 → $R : \{a,b,c,d\} \to \{0,1,2,3,4\}$

a) Yes

b) No

c) Yes

d) No.

Que 22

a) $L_n = \alpha^n + \beta^n$

b) $a_n = 2^n + (-2)^{n+1} + 2 \cdot 3^n - (-4)^n \quad n \geq 0$

d) $a_n^{(p)} = a n^2 2^n$

Que 31

4