The final answer is highlighted with color yellow answer
The work is presented with text color dark blue work

Question 3:

a. Solve Exercise 8.2.2, section b from the Discrete Math zyBook.

(b) $f(n) = n^3 + 3n^2 + 4$. Prove that $f = \Theta(n^3)$.

Ans:
To prove that $f = \Theta(n^3)$, we need to prove that $f = O(n^3)$ and $g = \Omega(n^3)$.
(1) Prove that $f = O(n^3)$, and we need to find a positive integer n_0 and a positive constant c such that $n^3 + 3n^2 + 4 \leq cn^3$ for every $n \geq n_0$.
Let $n_0 = 1$ and $n \geq n_0$, $n^3 + 3n^2 + 4 \leq n^3 + 3n^3 + 4n^3 = 8n^3$.
Therefore, $n^3 + 3n^2 + 4 \leq cn^3$ for every $n \geq n_0$ when $n_0 = 1$, $c = 8$.

(2) Prove that $f = \Omega(n^3)$, and we need to find a positive integer n_0 and a positive constant c such that $n^3 + 3n^2 + 4 \geq cn^3$ for every $n \geq n_0$.
Let $n_0 = 1$ and $n \geq n_0$, $n^3 + 3n^2 + 4 \geq n^3$.
Therefore, $n^3 + 3n^2 + 4 \geq cn^3$ for every $n \geq n_0$ when $n_0 = 1$, $c = 1$.

---

b. Solve Exercise 8.3.5, sections a-e from the Discrete Math zyBook
MysteryAlgorithm

```
Input: a1, a2,...,an
        n, the length of the sequence.
        p, a number.
Output: ??

i := 1
j := n

While (i < j)
    While (i < j and ai < p)
        i := i + 1
    End-while
    While (i < j and aj ≥ p)
        j := j - 1
    End-while
    If (i < j), swap ai and aj
End-while

Return( a1, a2,...,an )
```

(a) Describe in English how the sequence of numbers is changed by the algorithm. (Hint: try the algorithm out on a small list of positive and negative numbers with p = 0)

Ans:

At the end of the program:
Case1: If $j = 0$ at the end of the program, then all numbers are larger than or equal to p and the sequence doesn't change.

Case2: If $i = n + 1$ at the end of the program, then all numbers are less than p and the sequence doesn't change.

Case3: If $a_i \geq p$ and $a_j < p$, then a_i and a_j swap, and i is incremented or j is decremented until i < j. At the end of the program, $1 \leq j < i \leq n$ and the input sequence is divided into two parts: a_1 to a_j are less than p, and a_i to a_n are larger than or equal to p.

Ex: input sequence = 15, -4, 3, 17, -22, 0, 9   n = 7   p = 0   i = 1   j = 7
output sequence = -22, -4, 3, 17, 15, 0, 9   i =3, j = 2

(b) What is the total number of times that the lines "i := i + 1" or "j := j - 1" are executed on a sequence of length n? Does your answer depend on the actual values of the numbers in the sequence or just the length of the sequence? If so, describe the inputs that maximize and minimize the number of times the two lines are executed.

Because the outer while loop ends when i < j, which means the loop ends when i = x and j = x-1, $1 \leq x \leq n$. The number of times that the lines "i := i + 1" are executed is x - 1. The number of times that the lines "j := j - 1" are executed is n - (x - 1) = n - x + 1. Add them up = (x - 1) + (n - x + 1) = n.

Ans:
The total number of the times that "i := i + 1" and "j := j - 1" are executed are **n**. The total number of times that the lines "i := i + 1" or "j := j - 1" are executed **depends on the length n of the sequence**.

(c) What is the total number of times that the swap operation is executed? Does your answer depend on the actual values of the numbers in the sequence or just the length of the sequence? If so, describe the inputs that maximize and minimize the number of times the swap is executed.

Ans:
The total number of times of the execution of the swap operations are **from 0 to $\lfloor n / 2 \rfloor$**, and it **depends on the actual values of the numbers**.

Minimum - 0:
Case 1: If the input sequence with $a_1, a_2, \ldots, a_n$ is in ascending order ($a_1 < a_2 < \ldots < a_n$), or in descending order($a_1 > a_2 > \ldots > a_n$), or with a same value($a_1 = a_2 = \ldots = a_n$), then the number of times of swap is 0 no matter what value p is.

Case 2: If there is a x in the input sequence such that $1 \leq x \leq n$ and a_1 to a_x are less than p, and a_{x+1} to a_n are larger than or equal to p, then the number of times of swap is 0.

Ex: input = 1, 2, 4, 5, 6, 7, 8, 9 p = 3,
output = 1, 2, 4, 5, 6, 7, 8, 9 (swap 0 time)

Maximum - $\lfloor n / 2 \rfloor$:
If the input sequence with $a_1, a_2, \ldots, a_n$, a_1 to $a_{\lfloor n / 2 \rfloor}$ are larger than or equal to p and $a_{\lfloor n / 2 \rfloor + 1}$ to a_n are less than p, then the number of times of swap is $\lfloor n / 2 \rfloor$. In other words, if the two inner loops i increases only by 1 and j decreases only by 1 in every iteration and then a_i and a_j swap, the program has the maximum number of swap operations.
Ex: input = 10, 9, 8, 7, 1, 2, 3, 4, 5 p = 6 n = 9
output = 5, 4, 3, 2, 1, 7, 8, 9, 10 (swap 4 times: 10 and 5, 9 and 4, 8 and 3, 7 and 2)

(d) Give an asymptotic lower bound for the time complexity of the algorithm. Is it important to consider the worst-case input in determining an asymptotic lower bound (using Ω) on the time complexity of the algorithm? (Hint: argue that the number of swaps is at most the number of times that i is incremented or j is decremented).

Ans:
Since the condition to end the outer loop and the two inner loops is i < j, the inner while loops must run n times in total. Moreover, there is a constant number of c for the swap operation, and the swap operation runs most at $\lfloor n / 2 \rfloor$times. So the outer while loops runs at most $n + \lfloor n / 2 \rfloor$ times. There are at most a constant d number of operations performed before and after the outer loop. The algorithm runs at most $n + \lfloor n / 2 \rfloor + d$ times. Therefore the worst-case time complexity of the algorithm is **$\Omega(n)$**.

If the swap operation runs 0 times in the best case, then the while loops still run n + 0 + d times, the time complexity of the algorithm is still $\Omega(n)$.

We can find that no matter how many run times of the swap operation, the loop must run n times. Therefore, **the worst case input is not important** to determine the lower bound on the time complexity.

(e) Give a matching upper bound (using O-notation) for the time complexity of the algorithm.

Ans:
Since the condition to end the outer loop and the two inner loops is i < j, the inner while loops must run n times in total. Moreover, there is a constant number of c for the swap operation, and the swap operation runs most at $\lfloor n / 2 \rfloor$times. So the outer while loops runs at most $n + \lfloor n / 2 \rfloor$ times. There are at most a constant d number of operations performed before and after the outer loop. The algorithm runs at most $n + \lfloor n / 2 \rfloor + d$ times. Therefore the upper bound for time complexity of the algorithm is **$O(n)$**.

Question 4:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.1.1, sections b, c

Digits = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
Letters = { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z }
Special characters = { *, &, $, # }

(b) Strings of length 7, 8, or 9. Characters can be special characters, digits, or letters.

> Let D be the set of digits, L the set of letters, and S the set of special characters. The three sets are mutually disjoint, so the total number of characters is $|D \cup L \cup S| = |D| + |L| + |S| = 10 + 26 + 4 = 40$.
> Let a be the set of password with length 7, $|a| = 40^7$
> Let b be the set of password with length 8, $|b| = 40^8$
> Let c be the set of password with length 9, $|c| = 40^9$
> The three sets are mutually disjoint, so the total number of password is $|a \cup b \cup c| = |a| + |b| + |c| = 40^7 + 40^8 + 40^9$
>
> Ans: The number of passwords: $40^7 + 40^8 + 40^9$

(c) Strings of length 7, 8, or 9. Characters can be special characters, digits, or letters. The first character cannot be a letter.

> Let D be the set of digits, L the set of letters, and S the set of special characters. The three sets are mutually disjoint, so the total number of characters is $|D \cup L \cup S| = |D| + |L| + |S| = 10 + 26 + 4 = 40$.
> The first character cannot be a letter , so the first character is $|D \cup S| = |D| + |S| = 14$
> Let a be the set of password with length 7, $|a| = 14 \times 40^6$
> Let b be the set of password with length 8, $|b| = 14 \times 40^7$
> Let c be the set of password with length 9, $|c| = 14 \times 40^8$
> The three sets are mutually disjoint, so the total number of password is $|a \cup b \cup c| = |a| + |b| + |c| = 14 \times 40^6 + 14 \times 40^7 + 14 \times 40^8 = 14(40^6 + 40^7 + 40^8)$
>
> Ans: The number of passwords: $14 \times (40^6 + 40^7 + 40^8)$

---

b) Exercise 5.3.2, section a

(a) How many strings are there over the set {a, b, c} that have length 10 in which no two consecutive characters are the same? For example, the string "abcbcbabcb" would count and the strings "abbbcbabcb" and "aacbcbabcb" would not count.

> We set the string with length 10 as 10 spaces : _ _ _ _ _ _ _ _ _ _. We have 3 options at the first position: a, b, c. Because two consecutive characters are not allowed, we only have two options at the

next position, and we only have two options at the next position of the next position...and so on. Therefore, the options for every space are: 3 2 2 2 2 2 2 2 2 2. We multiply all of them = 3×2^9 = 1536.

Ans: The number of the valid strings: 3×2^9 = 1536

---

c) Exercise 5.3.3, sections b, c

License plate numbers in a certain state consists of seven characters. The first character is a digit (0 through 9). The next four characters are capital letters (A through Z) and the last two characters are digits. Therefore, a license plate number in this state can be any string of the form: Digit-Letter-Letter-Letter-Letter-Digit-Digit

(b) How many license plate numbers are possible if no digit appears more than once?

We set the license plate numbers as 7 spaces : _ _ _ _ _ _ _. No digit appears more than once, meaning that the first digit has 10 options, and the second digit has 9 options, and the third digit has 8 options.
Therefore, the options for every space are: 10 26 26 26 26 9 8. We multiply all of them = $10 \times 26 \times 26 \times 26 \times 26 \times 9 \times 8 = 720 \times 26^4$ = 329,022,720.

Ans: The number of the valid license plate numbers: 720×26^4 = 329,022,720.

(c) How many license plate numbers are possible if no digit or letter appears more than once?

We set the license plate numbers as 7 spaces : _ _ _ _ _ _ _. No digit appears more than once, meaning that the first digit has 10 options, and the second digit has 9 options, and the third digit has 8 options. With the same rules, the first letter has 26 options, and the second letter has 25 options..and so on.
Therefore, the options for every space are: 10 26 25 24 23 9 8. We multiply all of them = $10 \times 26 \times 25 \times 24 \times 23 \times 9 \times 8$ = 258,336,000

Ans: The number of the valid license plate numbers: $10 \times 26 \times 25 \times 24 \times 23 \times 9 \times 8$ = 258,336,000

---

d) Exercise 5.2.3, sections a, b

Let B = {0, 1}. B^n is the set of binary strings with n bits. Define the set E_n to be the set of binary strings with n bits that have an even number of 1's. Note that zero is an even number, so a string with zero 1's (i.e., a string that is all 0's) has an even number of 1's.

(a) Show a bijection between B^9 and E_{10}. Explain why your function is a bijection.

Ans: B^9 = {000000000, 000000001,...., 111111111}. E_{10} = {0000000000, …, 1111111111}. We define a function f: $B^9 \rightarrow E_{10}$, and show that the function is 1-to-1 and onto, which means bijection.

If $x \in B^9$, then $f(x)$ = x appends 1 at the end if x has the odd number of 1's; x appends 0 at the end if x has the even number of 1's.
Ex: f(000011100) = 0000111001, f(110011000) = 1100110000.

f is 1-to-1 because we know that every $x \in B^9$ are distinct, and appending 1 or 0 at the end of them makes f(x) also be distinct. That is, if $x,y \in B^9$ and $x \neq y$, then $f(x) \neq f(y)$.

f is onto because for any $z \in E_{10}$, we can drop the last bit of z to obtain x. And f(x) = z, which means for every $z \in E_{10}$, there is a $x \in B^9$ such that f(x) = z.

(b) What is $|E_{10}|$?

There is a bijection between B^9 and E_{10}, so $|B^9| = |E_{10}|$. We can calculate $|B^9|$ to find $|E_{10}|$. B^9 is the set of binary strings with 9 bits, so $|B^9| = 2^9$. Therefore, $|E_{10}| = |B^9| = 2^9 = 512$.

Ans: $|E_{10}| = 2^9 = 512$.

Question 5:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.4.2, sections a, b

At a certain university in the U.S., all phone numbers are 7-digits long and start with either 824 or 825.

(a) How many different phone numbers are possible?

We set the license plate numbers as 824 and 4 spaces : 8 2 4 _ _ _ _. Every space has 10 options (0~9). Therefore, the options for every space are: 8 2 4 <u>10</u> <u>10</u> <u>10</u> <u>10</u>. We multiply all of them = 10^4, then we obtain the number of phone numbers starting with 824. Because the set of starting with 824 and the set of starting with 825 are mutually disjoint, we use the sum rule. $10^4 + 10^4 = 2 \times 10^4 = 20000$

Ans: The number of the different phone numbers: $10^4 + 10^4 = 20000$

(b) How many different phone numbers are there in which the last four digits are all different?

We set the license plate numbers as 824 and 4 spaces : 8 2 4 _ _ _ _. Since the last four digits are all different, the different numbers are P(10, 4). P(10, 4) = 10! / 6! = $10 \times 9 \times 8 \times 7 = 5040$.
Because the set of starting with 824 and the set of starting with 825 are mutually disjoint, we use the sum rule. 5040 + 5040 = 10080

Ans: The number of the different phone numbers: P(10, 4) + P(10, 4) = 10080

---

b) Exercise 5.5.3, sections a-g

How many 10-bit strings are there subject to each of the following restrictions?

(a) No restrictions.

The number of possible 10-bit strings means every bit has two options (0 or 1), so there are $2^{10} = 1024$ different strings.

Ans: $2^{10} = 1024$

(b) The string starts with 001.

The number of possible 10-bit strings starting with 001 means the bits of the last 7 bits all have two options (0 or 1), so there are $2^7 = 128$ different strings.

Ans: $2^7 = 128$

(c) The string starts with 001 or 10.

Based on the question(b), the number of possible 10-bit strings starting with 001 is 128. The number of possible 10-bit strings starting with 10 means the bits of the last 8 bits all have two options (0 or 1), so there are $2^8 = 256$ different strings. Because the set of starting with 001 and the set of starting with 10 are mutually disjoint, we use the sum rule. 128 + 256 = 384

Ans: $2^7 + 2^8 = 384$

(d) The first two bits are the same as the last two bits.

Since the first two bits are the same as the last two bits, we don't need to consider the last two bit strings. Therefore, we only need to care about the first 8 bits. The bits of the first 8 bits all have two options(0 or 1), so there are $2^8 = 256$ different strings.

Ans: $2^8 = 256$

(e) The string has exactly six 0's.

We define a bijection from the set of 10-bit strings that has exactly six 0's to the 6-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Let set S be the set of 10-bit strings that has exactly six 0's and set T be the 6-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. f: S → T. f is obvious 1-to-1 and onto.

The diagram of f:

1 2 3 4 5 6 7 8 9 10
0 0 0 0 0 0 1 1 1 1 {1, 2, 3, 4, 5, 6}
0 0 0 0 0 1 **0** 1 1 1 {1, 2, 3, 4, 5, 7}
0 0 0 0 0 1 1 **0** 1 1 {1, 2, 3, 4, 5, 8}
....

|T| = C(10, 6) = 10! / 6!4! = 210. Because the mapping is bijection, |S| = |T| = 210.

Ans: C(10, 6) = 210

(f) The string has exactly six 0's and the first bit is 1.

Since the first bit is 1, we only need to consider the last 9 bits. We define a bijection from the set of 9-bit strings that has exactly six 0's to the 6-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9}. Let set S be the set of 10-bit strings that has exactly six 0's and set T be the 6-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9}. f: S → T. f is obvious 1-to-1 and onto.

The diagram of f:

1 2 3 4 5 6 7 8 9
0 0 0 0 0 0 1 1 1 {1, 2, 3, 4, 5, 6}
0 0 0 0 0 1 **0** 1 1 {1, 2, 3, 4, 5, 7}
0 0 0 0 0 1 1 **0** 1 {1, 2, 3, 4, 5, 8}
....

|T| = C(9, 6) = 9! / 6!3! = 84. Because the mapping is bijection, |S| = |T| = 84.

Ans: C(9,6) = 84

(g) There is exactly one 1 in the first half and exactly three 1's in the second half.

We define a bijection from the set of 10-bit strings that has exactly one 1 in the first half and exactly three 1's in the second half to the 4-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Let set S be the set of 10-bit strings that has exactly one 1 in the first half and exactly three 1's in the second half and set T be the 4-subsets of {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. f: S $\rightarrow$ T. f is obvious 1-to-1 and onto.

The diagram of f:
(Choose 1 bit to set as 1 from the first 5 bits, and choose three bits as 1 from the last 5 bits. Then we use the product rule to find the number of all possible outcomes)

1 2 3 4 5 | 6 7 8 9 10
1 0 0 0 0 | 1 1 1 0 0 {1, 6, 7, 8}
0 1 0 0 0 | 1 1 1 0 0 {2, 6, 7, 8}
0 0 1 0 0 | 1 1 1 0 0 {3, 6, 7, 8}
....

|T| = C(5, 1) × C(5, 3) = (5! / 4!1!) × (5! / 2!3!) = 5 × 10 = 50. Because the mapping is bijection, |S| = |T| = 50.

Ans: C(5, 1) × C(5, 3) = 50

---

c) Exercise 5.5.5, section a
(a) There are 30 boys and 35 girls that try out for a chorus. The choir director will select 10 girls and 10 boys from the children trying out. How many ways are there for the choir director to make his selection?

To choose 10 boys from the 30 boys, there are C(30, 10) ways. To choose 10 girls from the 35 girls, there are C(35, 10) ways. Moreover, both boys and girls are required, so we use the product rule. The total ways are C(30, 10) × C(35, 10)

Ans: C(30, 10) × C(35, 10)

d) Exercise 5.5.8, sections c-f

This question refers to a standard deck of playing cards. If you are unfamiliar with playing cards, there is an explanation in the "Probability of an event" section under the heading "Standard playing cards." A five-card hand is just a subset of 5 cards from a deck of 52 cards.

(c) How many five-card hands are made entirely of hearts and diamonds?

> There are 13 hearts and 13 diamonds. There are C(26, 5) = 26! / 21!5! = 65,780 ways to select a subset of 5 cards from the set of 26 cards.
> (**If at least one heart and at least one diamond are required, then we need to minus the selections of all 5 hearts and 5 diamonds. That is C(26, 5) - C(13, 5) - C(13, 5) = 63206**)
>
> Ans: C(26, 5) = 65,780

(d) How many five-card hands have four cards of the same rank?

> There are C(13, 1) ways to select a subset of 1 rank for the four cards from the set of 13 ranks. We don't need to select the suits for the four cards because the suits are fixed. Then we choose 1 card from the remaining 48 cards, and there are C(48, 1) ways. The four cards with the same rank and the last card must be selected, so the product rule is applied to obtain C(13, 1) $\times$ C(48, 1) = 13 $\times$ 48 = 624.
>
> Ans: C(13, 1) $\times$ C(48, 1) = 624

(e) A "full house" is a five-card hand that has two cards of the same rank and three cards of the same rank. For example, {queen of hearts, queen of spades, 8 of diamonds, 8 of spades, 8 of clubs}. How many five-card hands contain a full house?

> There are 13 ways to select the rank for the 3 cards with the same rank and 12 remaining ways to select the rank for the pair with the same rank. There are C(4, 3) ways to select a subset of 3 suits from the set of the set of 4 suits. There are C(4, 2) ways to select a subset of 2 suits from the set of the set of 4 suits. Then we apply the product rule to obtain the number of different full house, 13 $\times$ 12 $\times$ C(4, 3) $\times$ C(4, 2) = 13 $\times$ 12 $\times$ 4 $\times$ 6 = 3744
>
> Ans: 13 $\times$ 12 $\times$ C(4, 3) $\times$ C(4, 2) = 3744

(f) How many five-card hands do not have any two cards of the same rank?

> Five-card hands without any two cards of the same rank means every card with the different rank. There are C(13, 5) ways to select a subset of 5 ranks from the set of 13 ranks. Then for each rank, there are 4 ways to choose a suit for it. Therefore, there are C(13, 5) $\times$ 4^5 = (13! / 8!5!) $\times$ 1024 = 1287 $\times$ 1024 = 1,317,888
>
> Ans: C(13, 5) $\times$ 4^5 = 1,317,888

e) Exercise 5.6.6, sections a, b

A country has two political parties, the Demonstrators and the Repudiators. Suppose that the national senate consists of 100 members, 44 of which are Demonstrators and 56 of which are Repudiators.

(a) How many ways are there to select a committee of 10 senate members with the same number of Demonstrators and Repudiators?

> There are C(44, 5) ways to select a subset of 5 senate members from the set of the Demonstrators. There are C(56, 5) ways to select a subset of 5 senate members from the set of the Repudiators. The 5 Demonstrators and the 5 Demonstrators must be selected, so the product rule is applied, and there are C(44, 5) $\times$ C(56, 5) ways to select a committee of 10 senate members.
>
> Ans: C(44, 5) $\times$ C(56, 5)

(b) Suppose that each party must select a speaker and a vice speaker. How many ways are there for the two speakers and two vice speakers to be selected?

> There are 44 options for the Demonstrators to choose the speaker, and there are 43 options to choose the vice speaker. Moreover, There are 56 options for the Repudiators to choose the speaker, and there are 55 options to choose the vice speaker. Because each party must choose a speaker and a vice speaker, the product rule is applied. Therefore, there are $44 \times 43 \times 56 \times 55$ ways.
>
> Ans: $44 \times 43 \times 56 \times 55 = 5{,}827{,}360$

Question 6:

Solve the following questions from the Discrete Math zyBook:

a) Exercise 5.7.2, sections a, b

A 5-card hand is drawn from a deck of standard playing cards.

(a) How many 5-card hands have at least one club?

Let S be the set of selecting a subset of 5 cards from the set of 52 cards. Let C be the set of 5-card hands with at least one club. Let $\overline{C}$ be the set of 5-card hands without any club. We can use |S| - |$\overline{C}$| = |C| to find |C|. |S| = C(52,5). |$\overline{C}$| = C(39, 5). Therefore |C| = C(52, 5) - C(39, 5) = (52! / 47!5!) - (39! / 34!5!) = 2,598,960 - 575,757 = 2,023,203.

Ans: C(52, 5) - C(39, 5) = 2,023,203

(b) How many 5-card hands have at least two cards with the same rank?

Let S be the set of selecting a subset of 5 cards from the set of 52 cards. Let C be the set of 5-card hands with at least two cards with the same rank. Let $\overline{C}$ be the set of 5-card hands with five different ranks. We can use |S| - |$\overline{C}$| = |C| to find |C|. |S| = C(52,5). |$\overline{C}$| = C(13, 5) $\times$ 4^5. Therefore |C| = C(52, 5) - C(13, 5) $\times$ 4^5 = (52! / 47!5!) - (13! / 8!5!) $\times$ 1024 = 2,598,960 - 1287 $\times$ 1024 = 2,598,960 - 1,317,888 = 1,281,072.

Ans: C(52, 5) - C(13, 5) $\times$ 4^5= 1,281,072.

---

b) Exercise 5.8.4, sections a, b

20 different comic books will be distributed to five kids.

(a) How many ways are there to distribute the comic books if there are no restrictions on how many go to each kid (other than the fact that all 20 will be given out)?

When we distribute a book, we have 5 kids to choose. We distribute 20 times(there are 20 books), so we have 5^{20} ways to distribute the books.

Ans: 5^{20}

(b) How many ways are there to distribute the comic books if they are divided evenly so that 4 go to each kid?

C(20, 4) $\times$ C(16, 4) $\times$ C(12, 4) $\times$ C(8, 4) $\times$ C(4, 4) = 20! / 4!4!4!4!4!

Ans: $20! / 4!4!4!4!4!$

Question 7:

How many one-to-one functions are there from a set with five elements to sets with the following number of elements?

a) 4

If f: $S \to T$ and f is 1-to-1, then $|S| \leq |T|$. So there is no one-to-one function from a set with 5 elements to the set with 4 elements.

Ans: 0

b) 5

We assume that f: $S \to T$ and f is 1-to-1, and S = {1, 2, 3, 4, 5}, T = {a, b, c, d, e}. For 1, there are 5 options for it to map; for 2, there are 4 options for it to map....and so on. Therefore,the number of 1-to-1 functions are P(5, 5) = 5! = 120.

Ans: 5! = 120

c) 6

We assume that f: $S \to T$ and f is 1-to-1, and S = {1, 2, 3, 4, 5}, T = {a, b, c, d, e, f}. For 1, there are 6 options for it to map; for 2, there are 5 options for it to map....and so on. Therefore,the number of 1-to-1 functions are P(6, 5) = 6! / 1! = 6! = 720.

Ans: 6! = 720

d) 7

We assume that f: $S \to T$ and f is 1-to-1, and S = {1, 2, 3, 4, 5}, T = {a, b, c, d, e, f, g}. For 1, there are 7 options for it to map; for 2, there are 6 options for it to map....and so on. Therefore,the number of 1-to-1 functions are P(7, 5) = 7! / 2! = 2520.

Ans: 7! / 2! = 2520