Homework 7

Oscar Ramirez

Questions 3 - 7
NYU Tandon CS Extended Bridge Summer 2022
NetID: or2092

Question 3:

Exercise 8.2.2

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

Proof:
Assuming $f(n) = n^3 + 3n^2 + 4$ and $g(n) = n^3$, select $c = 1$ and $n_0 = 1$.
We will prove that for any $n \geq 1$, $f(n) \geq g(n)$
For $n \geq 1$, we know that $n \leq n^2 \leq n^3$.
So $n^3 \leq n^3 + 3n^2 + 4$.
$n^3 = g(n)$, so $g(n) \leq f(n)$. Alternatively, $f(n) \geq g(n)$.
Therefore, for all $n > n_0$, $f(n) \geq g(n)$ and $f = \Omega(g)$

Select $c = 8$, and $n_0 = 1$. We will show that for any $n \geq 1$, $f(n) \leq 8 \cdot g(n)$.
For $n \geq 1$, we know that $n \leq n^2 \leq n^3$.
So $n^3 + 3n^2 + 4 \leq n^3 + 3n^3 + 4 \leq n^3 + 3n^3 + 4n^3$.
$n^3 + 3n^3 + 4n^3 = 8n^3 = 8(g(n))$.
Therefore, for all $n > n_0$, $f(n) \leq 8g(n)$ and $f = O(g)$.

Finally,

$$1 \cdot g(n) \leq f(n) \leq 8 \cdot g(n), \text{ for all } n \geq 1$$

Since, $f = O(n^3)$ and $f = \Omega(n^3)$, therefore $f = \theta(n^3)$. ■

Exercise 8.3.5

a)
The algorithm is going through a sequence of numbers of length n from both ends. The variable 'i' is initialized as the first element of the sequence to start, while the variable 'j' is initialized as the last element of the sequence to start. In a loop, the variable 'i' is incremented by 1 until it finds a number in the sequence that is $\geq$ p, or until $i > j$, while the variable 'j' is decremented from n until it finds a number in the sequence that is $<$ p, or until $i > j$. The two numbers are then swapped if conditions are met. This loop repeats until $i \geq j$.

b)
The maximum number of times the lines "i := + 1" or "j := j - 1" are executed is $\frac{n}{2}$. The number of executions depends on the values in the sequence and how they relate to p. The length of the sequence maximizes the number of times the two lines are executed, while the value of p and the elements in the sequence minimize the number of times the two lines are executed.

c)
The total number of times the swap operation is executed depends on the actual values of the numbers in the sequence. The maximum number of times the swap is executed is $\frac{n}{2}$ times, while the minimum number of times the swap is executed is zero (the case that there are an odd number of elements, p is in the middle, and all numbers to the left are less than p, and all numbers to the right are more than p).

d)
An asymptotic lower bound for the time complexity of the algorithm would be $\Omega(n)$. Since the number of executions is at most $\frac{n}{2}$, there wouldn't be a need to consider the worst-case input at $\theta(n)$.

e)
A matching upper bound for the time complexity of the algorithm would be $O(n)$.

Question 4:

Exercise 5.1.2

b)

$$\mathbf{40^9 \cdot 40^8 \cdot 40^7}$$

The number of possible characters, counting digits, letters and special characters is 40. So we can use the sum and product rules to count find the total possible passwords of each length, and then add them together.

c)

$$\mathbf{14(40^8 + 40^7 + 40^6)}$$

The number of possible characters is 40. Given that the first cannot be a letter, the remaining options are digits and special characters (14 total). The remaining characters can be any of 40 characters giving us $14 \cdot 40^8 + 14 \cdot 40^7 + 14 \cdot 40^6$. Factoring out 14 gives the answer above.

Exercise 5.3.2

a)

$$\mathbf{3 \cdot 2^9}$$

We can find the total number of strings by counting the options of each character and using the product rule. The first character can be any of the elements in the set {a, b, c}, however the remaining characters in the string are limited to 2 options. An 'a' can only be followed by 'b' or 'c'. A 'b' can only be followed by 'a' or 'c'. And finally, 'c' can only be followed by 'a' or 'b'. So it would be $3 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2$, or $3 \cdot 2^9$.

Exercise 5.3.3

b)

$$\mathbf{10 \cdot 26^4 \cdot 9 \cdot 8}$$

For the sequence Digit-Letter-Letter-Letter-Letter-Digit-Digit, if no digit appears more than once then the options for digits are 10, 9, and 8 respectively. Since there are no restrictions for letters, the options are 26 for each instance. So $26^4 \cdot 10 \cdot 9 \cdot 8$ can be used to count the possible license plate numbers.

c)

$$\mathbf{10 \cdot 26 \cdot 25 \cdot 24 \cdot 23 \cdot 9 \cdot 8}$$

For the sequence Digit-Letter-Letter-Letter-Letter-Digit-Digit, if no digit appears more than once then the options for digits are 10, 9, and 8 respectively. If no letter can appear more than once, the options are 26, 25, 24, 23 respectively. So $26 \cdot 25 \cdot 24 \cdot 23 \cdot 10 \cdot 9 \cdot 8$ can be used to count the possible license plate numbers.

Exercise 5.2.3

a)

$$\mathbf{E_{10} \to B^9}$$

Given that E^n is the set of n-bit strings with an even number of 1's, B^n has a 2-to-1 correspondence to E^n. Alternatively $|E^n| = \frac{1}{2}|B^n|$. For every bit that we add onto the string, we are creating two possible strings. Either we're appending a 1, or we're appending a 0. In other words, by adding a bit, we're multiplying the total number of strings by two. However, we're only considering strings with an even number of 1's. This would mean that if there are an odd number of 1's, we're only considering the string with an added 1. And, if there are an even number of 1's, we're only considering the string with an added 0. In both cases, we're restricting the last bit to be the bit that ensures an even number of 1's, which would be shown as $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 1 = 2^9 \cdot 1$. So, $|E_{10}| = \frac{|B^{10}|}{2} = \frac{2^{10}}{2} = 2^9$. **Since $|\mathbf{E^{10}}| = |\mathbf{B^9}| = \mathbf{2^9}$, $\mathbf{E^{10}}$ and $\mathbf{B^9}$ are a bijection.**

b)

$$|E_{10}| = |B^9| = \mathbf{2^9}$$

We can think of the last digit in E as the digit that ensures there are an even number of 1's in the string. So, it is restricted to 1 option based on the number of 1's in the first n-1 digits. $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 1 = 2^9 \cdot 1$. We can also use the k-to-1 correspondence with B^{10}, so $2|E_{10}| = |B^{10}|$ or $|E_{10}| = \frac{|B^{10}|}{2} = \frac{2^{10}}{2} = 2^9$

Question 5:

Exercise 5.4.2

a)

$$\mathbf{2(10^4)}$$

Since the first 3 digits are predetermined, the total number of different phone numbers is equal to 10^4. Since there are 10 options for 4 digits. This goes for numbers that start with 824, and 825. So we can add them together $10^4 + 10^4 = 2(10^4)$

b)

$$\mathbf{2(10 \cdot 9 \cdot 8 \cdot 7)}$$

Since the first 3 digits are predetermined, and numbers cannot be repeated, the total number of different phone numbers is equal to $10 \cdot 9 \cdot 8 \cdot 7$. This goes for numbers that start with 824, and 825. So we can add them together $10 \cdot 9 \cdot 8 \cdot 7 + 10 \cdot 9 \cdot 8 \cdot 7 = 2(10 \cdot 9 \cdot 8 \cdot 7)$

Exercise 5.5.3

a)

$$\mathbf{2^{10}}$$

No restrictions, so a 10-bit string has 2 options for each bit. $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

b)

$$\mathbf{2^7}$$

The first 3 bits of the strings are predetermined to be 001. So for the remaining bits in each string, $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7$

c)

$$\mathbf{2^7 + 2^8}$$

For the strings starting with 001, $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7$. For the strings starting with 01, $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^8$. Then we can add the two for the total number.

d)

$$\mathbf{2^8}$$

If the first two digits, are the same as the last two digits, the result would have a 4-to-1 correspondence to a 6-bit-string. (e.g., xx222222xx, where xx = 00, 01, 10, 11). $4 \cdot 2^6 = 2^8$.

e)

$$\boldsymbol{\binom{10}{6}}$$

f)

$$\boldsymbol{\binom{9}{6}}$$

g)

$$\boldsymbol{\binom{5}{1}\binom{5}{3}}$$

Exercise 5.5.5

a)

$$\boldsymbol{\binom{30}{10}\binom{35}{10}}$$

Exercise 5.5.8

c)

$$\boldsymbol{\binom{26}{5}}$$

d)

$$\binom{13}{1}\binom{48}{1}$$

e)

$$\binom{13}{1}\binom{4}{2}\binom{12}{1}\binom{4}{3}$$

f)

$$\boldsymbol{\binom{13}{5}\cdot 4^5}$$

Exercise 5.6.6

a)

$$\binom{44}{5}\binom{56}{5}$$

b)

$$P(44, 2) \cdot P(56, 2)$$

Question 6:

Exercise 5.7.2

a)

$$\binom{52}{5} - \binom{39}{5}$$

b)

$$\binom{52}{5} - (\binom{13}{5} \cdot 4^5)$$

Exercise 5.8.4

a)

$$5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 = \mathbf{5^{20}}$$

b)

$$\binom{20}{4}\binom{16}{4}\binom{12}{4}\binom{8}{4}\binom{4}{4} = \frac{20! \cdot 16! \cdot 12! \cdot 8! \cdot 4!}{4!16! \cdot 4!12! \cdot 4!8! \cdot 4!4! \cdot 4!0!} = \frac{\mathbf{20!}}{\mathbf{4!4!4!4!4!}}$$

Question 7:

a)

$$\mathbf{0}$$

For a function to be one-to-one, the cardinality of the domain must be less than or equal to the co-domain. That is, n must be greater than or equal to k.

b)

$$P(5,5) = \frac{5!}{0!} = 5! = 120$$

c)

$$P(6,5) = \frac{6!}{1!} = 6! = 720$$

d)

$$P(7,5) = \frac{7!}{2!} = 2520$$