CSCE 222 Discrete Structures for Computing – Fall 2021
Hyunyoung Lee

Problem Set 8

> **Due dates:** Electronic submission of *yourLastName-yourFirstName-hw8.tex* and *yourLastName-yourFirstName-hw8.pdf* files of this homework is due on **Monday, 11/8/2021 11:59 p.m.** on `https://canvas.tamu.edu`. You will see two separate links to turn in the .tex file and the .pdf file separately. Please do not archive or compress the files. **If any of the two files are missing or unreadable, you will receive zero points for this homework.**

Name: Blake Lun **UIN: 131001178**

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Electronic signature: <u>Blake Lun</u>

Total 100 points. The problems are from the lecture notes posted on Perusall. Explain everything *in your own words*!

Problem 1. (20 points) Read Section 11.6 carefully before attempting this problem. Analyze the running time of the following algorithm using a step count analysis as shown in the Horner scheme (Example 11.40).

```
// search a key in an array a[1..n] of length n
search(a, n, key)          cost   times
  for k in (1..n) do        c1    [ n + 1 ]
    if a[k]=key then        c2    [ n ]
       return k             c3    [ 1 ]
  endfor                    c4    [ n ]
  return false              c5    [ 1 ]
```

(a) Fill in the `[  ]`s in the above code each with a number or an expression involving `n` that expresses the step count for the line of code.

(b) Determine the worst-case complexity of this algorithm and give it in the Θ notation. Show your work and explain using the definition of Θ involving the inequalities.

Solution. $T(n) = c1 * (n + 1) + c2 * (n) + c3 + c4 * (n) + c5 = \Theta(n)$ (following example 11.40)

Problem 2. (20 points) Read Section 11.6 carefully before attempting this problem. Analyze the running time of the following algorithm using a step count analysis as shown in the Horner scheme (Example 11.40).

```
// determine the number of digits of an integer num
binary_digits(num)          cost  times
  int cnt = 1                c1   [ 1 ]
  while (num > 1) do         c2   [ floor(log_2(n)) + 1 ]
    cnt = cnt + 1            c3   [ floor(log_2(n)) ]
    num = floor( num/2.0 )   c4   [ floor(log_2(n)) ]
  endwhile                   c5   [ floor(log_2(n)) ]
  return cnt                 c6   [ 1 ]
```

(a) Fill in the `[ ]`s in the above code each with a number or an expression involving `n` that expresses the step count for the line of code.

(b) Determine the worst-case complexity of this algorithm and give it in the Θ notation. Show your work and explain using the definition of Θ involving the inequalities.

Solution. $T(n) = c1 + c2 * (\lfloor log_2(n) \rfloor + 1) + c3 * \lfloor log_2(n) \rfloor + c4 * \lfloor log_2(n) \rfloor + c5 * \lfloor log_2(n) \rfloor + c6$ (following example 11.40)

Problem 3. (10 points) Section 12.1, Exercise 12.1. Specify what counting principle(s) you are using. Also explain how you got your final answer.

Solution. Let S_1 be the set of valid binary digits $\{0,\ 1\}$. There are only two valid binary options. The first digit has to be a 1 since if it was a 0, the digit wouldn't need to be there and it would just be a bit string of length 11. Using the multiplication principle, this comes out to $1 * 2^{11} = 2048$ different possibilities of bit strings with a length of 12.

Problem 4. (10 points) Section 12.1, Exercise 12.2. Specify what counting principle(s) you are using. Also explain how you got your final answer.

Solution. Let S_1 be the set of strings that holds each letter of the alphabet $S_1 = \{A, B, ..., Z\}$. Let S_2 be the set of digits $S_2 = \{0, 1, 2, ..., 9\}$. The license plate contains 3 alphabet letters and then four digits. Through the multiplication principle, the number of possibilities end up being $26^3 * 10^4 = 175,760,000$ different possibilities since there are 26 valid letters and 10 valid numbers.

Problem 5. (20 points) Section 12.1, Exercise 12.3. Specify what counting principle(s) you are using. Also explain *carefully* how you got your final answer.

Solution. S_1 is the set of lowercase alphabet letters being {a, b, c, ..., z}. S_2 is the set of decimal digits being {0, 1, 2, ..., 9}. The first character must be lowercase which limits the possibility to 26 options. The other characters can be either $\in S_1$ or $\in S_2$ or $S_1 S_2$; therefore, there are 36 possibilities for the remaining characters. The user has 3 choices for the length of their password: 6, 7, or 8 characters. By the multiplication principle, for 6 characters, the number of combinations is $26 * 36^5$. The 36 is to the 5th power since there are 5 remaining characters after the first one. For 7 characters, the number of combinations is $26 * 36^6$, and for 8 characters, the number of combinations is $26 * 36^7$. Using the addition principle, adding these possibilities together gives $(26 * 36^5) + (26 * 36^6) + (26 * 36^7) = 2.0956367278 * 10^{12}$ possibilities.

Problem 6. (20 points) Section 12.1, Exercise 12.5. Specify what counting principle(s) you are using. Also explain *carefully* how you got your final answer.

Solution. Let k^n denote all words of length n over an alphabet with k letters. When $n = 1$, there are k palindromes, k is simply k since it is just one letter length words. When $n = 2$, there are k palindromes, the two letters have to be the same so only one letter is used. When $n = 3$, there are k^2 palindromes since the ends of the word have to be the same letter and the letter in between can be anything. When $n = 4$. there are k^2 palindromes since the ends of the word have to be the same latter and the two letters in between must match. If we follow this pattern. We can see that it follows the pattern $k^{\lfloor \frac{n+1}{2} \rfloor}$. The floor function is added for when n is an even number. Finally, by the subtraction principle, we remove the palindromes from the set of all words which comes out to be $k^n - k^{\lfloor \frac{n+1}{2} \rfloor}$.

Checklist:

□ Did you type in your name and UIN?
□ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted.)
□ Did you sign that you followed the Aggie Honor Code?
□ Did you solve all problems?
□ Did you submit the .tex and .pdf files of your homework to the correct link on Canvas?