COMP 250, Winter 2017 - Homework 2

Mathieu Blanchette

February 2, 2017

This homework is due in whole on February 15th 2017, 23:59 Submit your solution to questions 1-8 as a PDF file. Submit your solution to question 9

Question 1 (20 points)

The algorithm below, called bubbleSort, sorts in increasing order the elements of an array.
Algorithm bubbleSort(A,n)
Input: An array $A[0...n-1]$ of n elements
Output: The array A is sorted
for $i \leftarrow 1$ **to** $n-1$ **do**
. **for** $j \leftarrow 0$ **to** $n-1-i$ **do**
. **if** $(A[j+1] < A[j])$ **then** /* swap A[j+1] and A[j] */
. $tmp \leftarrow A[j]$
. $A[j] \leftarrow A[j+1]$
. $A[j+1] \leftarrow tmp$

a) (5 points) What kind of input will yield the worst possible running time, for any fixed size n? Why?

b) (10 points) Let $T(n)$ be the number of primitive operations performed in this worst case. Write, for each line of the pseudocode, the number of primitive operations executed and the type (assignment, condition, arithmetic...) of each. Sum up everything to obtain $T(n)$, assuming all primitive operations take the same time. If $T(n)$ contains summations, replace them by their explicit value. Show your work step by step like we did in class.

c) (5 points) Give the simplest and most accurate big-Oh representation for $T(n)$.

Question 2. Running time analysis (8 points)

For each of the algorithms below, indicate the running time using the simplest and most accurate big-Oh notation. Assume that all arithmetic operations can be done in constant time. The first algorithm is an example. No justifications are needed.

Algorithm	**Running time in big-Oh notation**
Algorithm Example(n) $x \leftarrow 0$ **for** $i \leftarrow 1$ **to** n **do** . $x \leftarrow x + 1$	**O(n)**
Algorithm algo1(n) $i \leftarrow 1$ **while** $i < n$ **do** . $i \leftarrow i + 100$	
Algorithm algo2(n) $x \leftarrow 0$ **for** $i \leftarrow 1$ **to** n **do** . **for** $j \leftarrow 1$ **to** i **do** . $x \leftarrow x + 1$	
Algorithm algo3(n) $i \leftarrow n$ **while** $(i > 1)$ **do** . $i \leftarrow i/2$	
Algorithm algo4(n) $k \leftarrow 1$ **for** $i \leftarrow 1$ **to** 1000 . **for** $j \leftarrow 1$ to i . $k \leftarrow (k + i - j) * (2 + i + j)$	

Question 3 (10 points)

Prove by induction that $4^n < n!$ for any $n \geq 9$.
Note: $n! = n \cdot (n-1) \cdot (n-2) \cdot ... \cdot 2 \cdot 1$.

Question 4 (10 points)

Let

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3 \cdot T(n-1) + 2 & \text{if } n > 1 \end{cases}$$

Obtain an explicit formula for $T(n)$.

Question 5 (8 points)

Prove, using only the definition of $O()$, that $25n + 5$ is $O(n)$. Write your proof using proper mathematical formalism, as done in the lecture notes

Question 6 (8 points)

Prove using only the definition of $O()$ that $(n+10)^{2.5} + n^2 + 1$ is $O(n^{2.5})$. Write your proof using proper mathematical formalism, as done in the lecture notes.

Question 7 (8 points)

Prove, using only the definition of O(), that $(n + 1)^2$ is not $O(n)$. Write your proof using proper mathematical formalism, as done in the lecture notes.

Question 8 (8 points)

There exists a remote island with a certain number of inhabitants, 10 of whom have a blue face. For reasons that are easily understandable, having a blue face is a topic that is taboo on the island. There are no mirrors on the island and nobody is allowed to tell another person that this person has a blue face. In fact, their law specifies that if a person was to learn that he/she has a blue face, he/she would have to commit suicide at midnight that day. Since the topic is taboo, there is no way for a person to realize that he/she have a blue face, so everybody lives happily. Until...

A foreigner comes to visit the island. Before leaving the island, he announces to everybody: "There are some people on this island who have a blue face! You should all start thinking about it now." Ten days later (that is, on the the tenth midnight after this declaration), all ten blue-faced inhabitants commit suicide.

Question: How did they realize that they had a blue face? Give your answer as concisely as possible. You shouldn't need more than 10 lines to write your answer. Hint: All inhabitants of the island took CS250 and got an 'A' in it.

Question 9 (20 points)

In this question, you are asked to implement in Java a program that will help you cheat next time you play Scrabble. Use the code available at

`http://www.cs.mcgill.ca/~blanchem/250/hw2/Scrabble.java`

and the accompanying file

`http://www.cs.mcgill.ca/~blanchem/250/hw2/englishDictionary.txt`

In Scrabble, you are given a set of n characters (possibly with some letters present more than once) and have to assemble English words out of these letters. You want a program that will find and return the longest English word that can be made out the given letters (without using any of the n letters more than once, but allowing to use certain letters zero time). For example, given the letters aajpv, your method should return the String java. Given the letters bpocuretm, it should return computer. If there are more than one valid words of maximal length, then your method should return any one of them.

Hint: This can be done surprisingly simply and elegantly using a recursive algorithm (my code is 12 lines long). Here is how. We will generate every combination of letters and check if it is in the dictionary using the "contains" method of the myDictionary object (see example in code).

Your method will have the following structure:
public static String longestWord(char avail[], String prefix)

The method will return the longest valid English word that can be obtained by starting with the word prefix and adding any subset of the letters in array avail. For example,
longestWord(['a','e','r','t'], "compu") should return "computer".
longestWord(['x','z'], "compu") should return "".
longestWord(['b','p','o','c','u','r','e','t','m'], "") should return "computer"

The method will enumerate all words that start with the string prefix in order to find the longest valid extension. This will be done recursively, using the following principle: The longest valid extension of the word prefix is either

1. The empty string ""
2. prefix itself, if prefix is in the dictionary
3. the longest word that can be made starting with the word prefix + avail[0], completed with the letters avail[1],...,avail[avail.length-1]
4. the longest word that can be made starting with the word prefix + avail[1], completed with the letters avail[0],avail[2],...,avail[avail.length-1] ...
5. the longest word that can be made starting with the word prefix + avail[avail.length-1], completed with the letters avail[0],avail[1],...,avail[avail.length-2]