Project 2

Spring 2025 CPSC 335 - Algorithm Engineering

Due date: 03/28/2025 (11:59 pm)

Abstract

In this project, you will devise a solution, develop a pseudocode, and implement an algorithm for solving:

1. Array pattern and substrings Problem
2. A string analysis and encoding problem
3. The merging of sorted arrays, using Heap Sort or other algorithms

Algorithm 1: Target Terms or Substrings

Assume that you are given two arrays of items. The first array is a single concatenation of some notable cities in California. The second array contains names of some real or imaginary cities, that may be present in the first array.

Sample:
Input: A = ["thismetoaklandrialtofullertonmarcolongchinofresnovallejoclovissimithound"]
B = ['marco', 'clovis', 'rialto', 'oakland']

Output: Output_order = [7, 14, 29, 56]
Output_array =['oakland', 'riato', 'marco', 'clovis']

Your output should include an array containing starting indices of the target words in array one. In the example above, oakland is the first word, and can be found at index array_A[7]. All words are in small letters.

To Do:

a. Develop a complete and clear pseudocode for an algorithm to solve this problem. Your algorithm should return the indices of the target words (contents of array B) in ascending order of their appearance in array A. It should also print the words, according to the resulting order of appearance.
b. Mathematically analyze your pseudocode and state the efficiency class.
c. Implement your algorithm in either Python or C++.

d. Using the given sample files (in2a.txt), print resulting lists for the 3 input arrays. Your program should read-in the files; not manually hardcoded into the functions. Include them in your PDF report.

Algorithm 2: Run Encoding Problem

The ***String Run Encoding*** problem involves compressing a string to ensure that it takes less space. A *"run"* is a substring of repeated characters, for example "aaaa". Run-length encoding means replacing all similar characters with a single copy of the character and a count of how many times it appears. For example, "aaaa" is replaced with "4a", because there are 4 copies of the letter 'a'. In our run-length encoding problem, all runs are encoded, and the input strings are limited to lower-case letters and space characters. The string run encoding problem can be defined as follows:

String run encoding problem
input: a string S of n characters, where each character is a lower-case letter or space
output: a string C where each run of k repetitions of the character x is replaced with the string "kx"

Sample inputs and outputs:

- "ddd" becomes "3d"
- "heloooooooo there" becomes "hel8o there"
- "choosemeeky and tuition-free" becomes "ch2osem2eky and tuition-fr2e"

To Do:

e. Develop a complete and clear pseudocode for an algorithm to solve this problem.
f. Mathematically analyze your pseudocode and state the efficiency class. Step counts can be used.
g. Implement your algorithm in either Python or C++.
h. Using the given sample files above, print 3 resulting encoding, according to your implementation. Include them in your PDF report.

Algorithm 3: Merging Techniques

You are provided with grades of students from various sections of CPSC 120. The grading system allows the use of negative points. The grades appear to be sorted: the first item in each array is the least element in the respective array. We have decided to merge the various

lists into a single sorted array. Sample input arrays and the desired output are presented below:
Sample input:

```
all_lists =[ [2, 5, 9, 21],
             [-1, 0, 2],
             [-10, 81, 121]
             [4, 6, 12, 20, 150] ]
```

Sample output:

```
[ -10, -1, 0, 2, 2, 4, 5, 6, 9, 12, 20, 21, 81, 121, 150]
```

There are different ways of merging these lists. Given that the first element of each array is the smallest integer, you can build a list of smallest items. Pick the smallest out of the list of all smallest items. This will become the first item of the merged sorted list. You may then proceed to check all items in parent array. Another method of achieving this is through the use of Heap Sort. A Min heap can be used to store the smallest elements at any point in your algorithm.

To Do:

i. Develop a complete and clear pseudocode to merge the arrays
j. Your pseudocode should be able to merge any number of arrays, not necessarily 4.
k. Mathematically analyze your pseudocode and state the efficiency class
l. Implement your algorithm in either Python or C++
m. Using the given sample files (in2c.txt), print resulting merged lists for the 3 input arrays. Include them in your PDF report

General To Do list

1. Produce a brief written project report ***in PDF format***. Your report should include:
    a. Your name(s), CSUF-supplied email address(es), and an indication that the submission is for project 2.
2. Develop the pseudocodes and implement your algorithms in Python or C++
3. Produce a readme file (can be in txt), describing how to execute your program.
4. Submit all files, including your PDF report and codes, separately.

Grading Rubric

The suggested grading rubric is given below:

Algorithm 1 (30 points)

a. Clear and complete Pseudocode = 5 points
b. Mathematical analysis and correct Big *O* efficiency class = 5 points
c. Successful compilation of codes= 10 points
d. Produces accurate results = 10 points

Algorithm 2 and 3 (35 points each)

e. Clear and complete Pseudocode = 5 points
f. Mathematical analysis and correct Big *O* efficiency class = 5 points
g. Successful compilation of codes= 15 points
h. Produces accurate results (3 output lists for Algo 2 and 3) = 10 points

Submitting your code

Submit your files to the Project 2 Assignment on Canvas. It allows for multiple submissions. You may submit your files **separately**. Do not zip or use .rar.

Ensure your submissions are your own works. Be advised that your submissions may be checked for plagiarism using automated tools. Do not plagiarize. As stated in the syllabus, a submission that involves academic dishonesty will receive a 0% score on that assignment. A repeat offense will result in an "F" in the class and the incident will be reported to the Office of Student Conduct.

Deadline

The project deadline is **Friday, March 28, by 11:59 pm** on Canvas.
Penalty for late submission (within 48 hours) is as stated in the syllabus. Projects submitted more than 48 hours after the deadline will not be accepted.