Project 3: Array Sorting and Merging

Spring 2022 CPSC 335 - Algorithm Engineering
Instructor: Dr. Sampson Akwafuo

Abstract

In this project you will implement

1. An array sorting problem, using a specified pattern
2. Merging of sorted arrays, using Heap Sort or other algorithms

Algorithm 1: Finding Target Substrings

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

Sample:

Input: Array_A = ["hemetoaklandrialtofullertonmarcolongchinofresnovallejoclovissimithousand"]
Array_B = ['clovis', 'vallejo', 'rialto', 'marco']

Output: Output_order =[12, 27, 47, 54]
Output_array =['rialto', 'marco', 'vallejo', 'clovis']

Your output should be an array containing starting indices of the target words in array one. In the example above, rialto is the first word, and can be found at index array_A[12]. 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 appearances in Array_A. It should also print the words, according to the resulting 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 (in3AS.txt), print resulting lists for the 3 input arrays. Include them in your PDF report.

Algorithm 2: 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. You decide 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, 1, 2, 2, 4, 5, 6, 9, 12, 20, 21, 81, 121, 150]
```

There are different ways of merging the given 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:

a. Develop a complete and clear pseudocode to merge the arrays
b. Your pseudocode should be able to merge any number of arrays, not necessarily 4.
a. Mathematically analyze your pseudocode and state the efficiency class
b. Implement your algorithm in either Python or C++
c. Using the given sample files (in3B.txt), print resulting merged lists for the 3 input arrays. Include them in your PDF report

To Do

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 3.
2. Develop the pseudocodes and implement your algorithms in Python or C++
3. Produce a readme file, describing how to execute your program.
4. Submit all files, including your codes separately. (Do not zip).

Grading Rubric

The suggested grading rubric is below.

Algorithm 1 and 2 (50 each)

a. Clear and complete Pseudocode = 10 points
b. Mathematical analysis and correct Big *O* efficiency class = 5 points
c. Successful compilation of codes= 20 points
d. Produces accurate results (3 output lists) = 15 points

Submitting your code

Submit your files to the Project 3 Assignment on Canvas. It allows for multiple submissions. You can submit your files separately or as a zip folder (do not user .rar)

Ensure your submissions are your own works. Be advised that your submissions may be checked for plagiarism using automated tools.

Deadline

The project deadline is on **Friday, April 22, 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.