CSC311: The Design and Analysis of Algorithms

Computer Science Department — Second Semester 1443

King Saud University — Project

---

*This project is divided into two parts, and carries **15%** of your grade.*

Due:

- **Phase 1:** Thursday, March 10, 2022, at 11:00 P.M.
- **Phase 2:** Thursday, March 31, 2022, at 11:00 P.M.

Phase 1: The median of two sorted arrays

Assume you have two sorted arrays of equal size (n each), write a program to find the median in an array of size $2n$, resulting from merging the two earlier arrays.
Note that the median of a sorted numbers is the middle one in case of odd n and the average of the middle two numbers when n is even.
After merging both arrays, the size of the larger array will be $2n$ i.e. an even value.
For this project let's assume n is odd.
Solve the problem using:

1. Brute Force BF approach and provide the time and space complexity of your solution. How to enhance it?
2. Divide and Conquer D&C approach and provide the time and space complexity of your solution. Then, set up a recurrence relation for the algorithms basic operation and solve it.

Phase 2: Text formating

In a software systems for document preparation such as MS Word or Latex, one routinely encounters the balanced lines, with reference to the number of characters that can be put in one line. Where we want to balance empty spaces in lines so we do not have some lines with lots of extra spaces and other lines with small amount of extra spaces.
The input text is a sequence of n words of lengths $l_1, l_2, ..., l_n$. Assume that the length of each word is smaller than the line width, M. The goal is to print this paragraph in a number of lines that hold a maximum of M characters each while minimizing the sum, over all lines (except the last), of the cubes of the numbers of extra space characters at the ends of each line, which is calculated as follow:
Given a line with capacity of M characters, and contains words i through j, where $i \leq j$

$$M - j + i - \sum_{k=i}^{j} l_k$$

Implement a Dynamic Programming algorithm to print an optimized paragraph then analyze the efficiency of your algorithm.

Illustration Example

Consider the following text:
CSC311 The Design and Analysis of Algorithms
with $M = 13$ *(M refers to the line width)*
below are two possible formating with different costs

C	S	C	3	1	1							
T	h	e		D	e	s	i	g	n			
a	n	d		A	n	a	l	y	s	i	s	
o	f		A	l	g	o	r	i	t	h	m	s

Table 1: format 1: Cost=$7^3 + 3^3 + 1^3 = 371$, (Not optimal)

C	S	C	3	1	1		T	h	e			
D	e	s	i	g	n		a	n	d			
A	n	a	l	y	s	i	s		o	f		
A	l	g	o	r	i	t	h	m	s			

Table 2: format 2: Cost=$3^3 + 3^3 + 2^3 = 62$

Deliverables

- The source code (the project folder). Make sure your code is well organized and documented. You may use any language you like, Java, Python, C++ ... etc
- A report that includes the following sections:
  - Cover page
  - **Phase 1**:
    1. A description of the Brute force algorithm (pseudo-code with explanation)
    2. What is the time and space complexity of your BF solution ?
    3. Enhancement if possible
    4. A description of the Divide and conquer algorithm (pseudo-code with explanation)
    5. What is the time and space complexity of your D&C solution?
    6. What is the recurrence relation for the algorithms basic operation ? and the solution.
    7. Sample run on the provided cases
    8. Source code (in Courier New with font size:8) with comments
    9. Describe challenges faced and how you tackled them
  - Phase 2:
    1. A description of the algorithm (pseudo-code with explanation)
    2. Time and space complexity
    3. Sample run
    4. Source code (in Courier New with font size:8) with comments

5. Describe challenges faced and how you tackled them
6. Student peer evaluation form (see Team Work Evaluation table below).

Evaluation Rubric

Evaluation rubric is divided into two parts: First part evaluates Team Work and second part addresses the functional requirements. The code in grade assignment is shown to the right.

Code	
Fully satisfied	1
Partially satisfied	0.5
Not satisfied	0

Part 1: Team Work			
Criteria	**Student 1**	**Student 2**	**Student 3**
Work division: Contributed equally to the work			
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA			
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions			
Time management: Attending on time, being ready to start the demo, good time management in discussion and demo.			
Total/2			

Part 2: Functional Requirements		
	Criteria	**Evaluation**
General	Overall quality of the code implementation (organization, clearness, design,...)	
	Complete report with well organized sections	
	Total/2	
Phase 1	Brute force algorithm description	
	BF time and space complexity and enhancement	
	Divide and conquer algorithm description	
	D&C time and space complexity	
	Implementation correctness + sample run + recurrence relation	
	Total/5.5	
Phase 2	Algorithm description (pseudo-code with explanation)	
	Time and space complexity	
	Implementation correctness + sample run	
	Total/5.5	

Important notes:

- No late submissions accepted.
- The group leader is asked to upload everything on LMS as a compressed file (zip) containing (source code and report).

- Discussions will be conducted per phase, within one week of its submission.
  - You need to prepare a simple presentation for the discussion, include:
    * The pseudo-code that you will explain.
    * Sample run
    * challenges faced and how you tackled them.

Good luck and have fun!

Input Sample Phase 1

example 1
Input: $A = [1, 3, 6], B = [2, 8, 12]$

Output: 4.5

when merging the two sorted arrays, the result is a sorted array of size 2n=6 contains [1, 2, 3, 6, 8, 12]. The median of an even array size is the average of the two middle elements at index 2 and 3 i.e. (3 + 6)/2 = 4.5

example 2
Input: $A = [1, 2, 3], B = [4, 5, 6]$

Output: 3.5

when merging the two sorted arrays, the result is a sorted array of size 2n=6 contains [1, 2, 3, 4, 5, 6]. The median of an even array size is the average of the two middle elements at index 2 and 3 i.e. (3 + 4)/2 = 3.5

example 3
Input: $A = [1, 3, 4, 6, 9], B = [2, 5, 7, 8, 10]$
Output: 5.5

when merging the two sorted arrays, the result is a sorted array of size 2n=10 contains [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The median of an even array size is the average of the two middle elements at index 4 and 5 i.e. (5 + 6)/2 = 5.5