Indian Institute of Technology Ropar

Data Structures and Algorithms

CS506 and CS205

Lab Assignment 2

DEADLINE: 30th August 2024; 11:59PM

Programming Assignment:

The purpose of this assignment is to apply multiple concepts at the same time to make an efficient sorting algorithm.

(A) Merge Sort using Linked List

Problem Statement:
You need to implement Merge Sort using Linked List which will be In-Place Sorting. You need to write Linked List code and a function for adding elements into the linked list (Add in the end). Please refer to the page to learn about the In-Place algorithm.

Programming Language:
C/C++ Programming (Any kind of inbuilt libraries/functions can't be used. You can't use any container even. ***Otherwise penalty will be imposed***)

Expected Documents:

1. Program file

Instruction for Programming:

In the first line, the user needs to give the number of test cases (T) as input.
Now for each test case, there will be two lines for input:
The first line should be the size of the linked list (N)
In the second line read the elements and build the linked list parallelly.

Constraints:
Time Complexity: O(N*logN)

Space Complexity: O(1) *(If implementing in-place sorting)*

Example:

2
5
1 5 9 6 7
3
20 30 15

Expected Output:

1 5 6 7 9
15 20 30

Explanation:

2 -> 2 Test Cases
5 -> Length of the linked list in Test Case 1
1 5 9 6 7 -> Linked list of Test Case 1
3 -> Length of the linked list in Test Case 2
20 30 15 -> Linked list of Test Case 2

(B) Quick Sort using Median of Median:

Problem Statement:

Implement and analyze the Quick Sort algorithm using the Median of Medians technique, starting with a subarray size of 5. Extend the implementation to support different subarray sizes (specifically 3,5,7) and compare the performance of these variants against the standard Quick Sort algorithm (which does not use the Median of Medians for pivot selection). Evaluate the algorithms based on execution time and the number of comparisons. Document the results, including graphical representations of the performance differences, and summarize the impact of different subarray sizes on Quick Sort's efficiency.

Programming Language:
C/C++ Programming (Any kind of inbuilt libraries/functions can't be used. You can't use any container even. ***Otherwise penalty will be imposed***)

Expected Documents:

1. Program file (Only with subarray of size 5)
2. Analytical Report

Instruction for Programming:

In the first line, the user needs to give the number of test cases (T) as input.
Now for each test case, there will be two lines for input:
The first line should be the size of the array (N)
The second line should be the N elements in the array.

Example:

```
2
5
1 5 9 6 7
3
20 30 15
```

Expected Output:

```
1 5 6 7 9
15 20 30
```

Explanation:

2 -> 2 Test Cases
5 -> Length of the array in Test Case 1
1 5 9 6 7 -> Array of Test Case 1
3 -> Length of the array in Test Case 2
20 30 15 -> Array of Test Case 2

(C) Heap Sort

Problem Statement:
You need to implement a Heap Sort algorithm.

Programming Language:
C/C++ Programming (Any kind of inbuilt libraries/functions can't be used. You can't use any container even. ***Otherwise penalty will be imposed***)

Expected Documents:

1. Program file

Instruction for Programming:

In the first line, the user needs to give the number of test cases (T) as input.
Now for each test case, there will be two lines for input:
The first line should be the size of the array (N)
The second line should be the N elements in the array.

Example:

2
5
1 5 9 6 7
3
20 30 15

Expected Output:

1 5 6 7 9
15 20 30

Explanation:

2 -> 2 Test Cases
5 -> Length of the array in Test Case 1
1 5 9 6 7 -> Array of Test Case 1
3 -> Length of the array in Test Case 2
20 30 15 -> Array of Test Case 2

Instructions for Programming:

1. Don't put unnecessary pieces of information or codes.
2. Good documentation in the codes is highly appreciated. (You may use comments)
3. Please remember to make modular codes.

Naming conventions of Files:

FirstName_EntryNo_CS506_Lab02_ProgA.c (For the Merge Sort)
FirstName_EntryNo_CS506_Lab02_ProgB.c (For the Quick Sort)
FirstName_EntryNo_CS506_Lab02_ProgC.c (For the Heap Sort)
FirstName_EntryNo_CS506_Lab02_ReportB.pdf (For the report of Quick Sort)

**Students of CS205 will write CS205 instead of CS506*

Example:-
Rejoy_2023CSM1011_CS506_Lab02_ProgA.c
Rejoy_2023CSM1011_CS506_Lab02_ProgB.c
Rejoy_2023CSM1011_CS506_Lab02_ProgC.c
Rejoy_2023CSM1011_CS506_Lab02_ReportB.pdf

Regarding Plagiarism Check:

Do not copy from each other or any source on the internet. Plagiarism will be checked. If any plagiarism is found (above a reasonable threshold), the student will be given heavy penalties (which may even lead to an F grade in the course).

For any clarification, you may contact the TAs in the lab sessions:

Shradha Sharma , *Madhav Mishra* , *Rejoy Chakraborty*

Indian Institute of Technology

Data Structure and Algorithms

CS506 and CS205

Problem Set (Associated with Lab 2)

Deadline: 30th August 2024

Questions

Question 1

For each of the following statements, decide whether it is always true, never true, or sometimes true for asymptotically nonnegative functions f and g. If it is always true or never true, explain why. If it is sometimes true, give one example for which it is true, and one for which it is false.

1. $f(n) = O(f(n)^2)$
2. $f(n) + g(n) = \Theta(\max(f(n), g(n)))$
3. $f(n) + O(f(n)) = \Theta(f(n))$
4. $f(n) = \Omega(g(n))$ and $f(n) = o(g(n))$
5. $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$.

Question 2

For each of the following recurrences, give an expression for the runtime $T(n)$ if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

1. $T(n) = T(n/2) + 2^n$
2. $T(n) = 2^n T(n/2) + n^n$
3. $T(n) = 2T(n/2) + n \log n$
4. $T(n) = 2T(n/2) + \frac{n}{\log n}$
5. $T(n) = 64T(n/8) - n^2 \log n$

Instructions for Problem Set

- Submit handwritten solutions mentioning your name, entry number, course code (CS205/CS506).
- Discussion with other friends or taking help from any other sources is acceptable. But you need to mention proper citation/reference from the source and mention the names with whom you have discussed. Otherwise, marks will be deducted.
- You need to submit the hardcopy of the answer script to the TAs in Lab 201/207/314 on/before 30th August at 6 pm.
- It is advisable to upload a softcopy too in the Google Classroom. Though it's not mandatory.
  Softcopy filename should be: FirstName_EntryNo_CS506_Lab02_ProbSet.pdf
  (CS205 students will write CS205 instead of CS506)
  For Example: Rejoy_2023CSM1011_CS506_Lab02_ProbSet.pdf

Contact TAs

Please feel free to contact us if you have any queries.

- Shradha Sharma
- Madhav Mishra
- Rejoy Chakraborty