Design and Analysis of Algorithms

Assignment-01

Dr. Ram P Rustagi
Sem IV (2019-H1)
Dept of CSE, KSIT/KSSEM
rprustagi@ksit.edu.in

Assignment Logistics

- Make a group of up to 3 team members.
  – Can be 2 as well, Individual is not recommended.
  – Any team member can be asked to explain
- All assignments are to be submitted online in github.
  – Program (java/C/C++/python) should run on Linux.
- 1 day late submission: 25% penalty
- 2 days late submission: 50% penalty.
- Early submission may yield bonus marks
  - Early submission by 2 days: 25% bonus marks.
- Expectation: Using brute force techniques.
  - Program should also output basic number of operation carried out e.g. comparison, Add/multiply etc.

Assignment Logistics

- There are total of 12 programming questions.
- You can choose any programming language
- Each group/team will be assigned one of the questions and need to submit the same question.
- A team is encouraged to do other non-assigned questions to help improve learning.
  - Team may be given bonus marks (as per the discretion of the instructor). The bonus marks may count towards future assignments.
- Plagiarism (copy) will result in 0 marks for all
- Any partial code from net (googling) should be cited with URL along with explanation

Assignment Submission Details

- Each submission (on github) should include
  - The program e.g. `Asn01P01.java/C/Cpp/py ...`
  - `Readme.txt`: should contain
    1. Team details (Names, USN)
    2. Contribution of each team member
    3. Instructions to run the program
    4. Details on example invocation and output
    5. Challenges faced and how did you address these
    6. What did you learn from this assignment
  - `Output.txt`
    1. output of program with your sample data
    2. Total number of basic (key) operations i.e. a computation of time complexity.

Q01: Find Missing Number

- Given input file containing `N-1` distinct numbers between 1 and N (both inclusive) in random order, find the missing number. Each number is on one line. You are not allowed to use any data structures that stores all of the number in one place i.e. you can only use O(1) memory space (data structures) but not O(fn(N)), e.g. O(log N), O(N). Neither you are supposed to use any in-built sort or other library function.
  - Note: Any built in sort function uses O(N) memory space, and thus not permitted.
  - You can use any number of files, and each file size can be O(N). All such created files should be removed when program execution is complete.

Q02: Hanoi Towers with 4 towers

- Implement Hanoi's tower using 4 towers to transfer N discs from tower 1 to tower 2 using tower 3 and 4 as temporary holding place. Compute the time complexity (i.e. total number of disc moves). The program should output each tower status after each move.
  - For example, for 4 discs, and 3 towers, tower status after each move should be like below

Initial	Move 1	Move 2	Move 3
['D1', 'D2', 'D3', 'D4']	['D2', 'D3', 'D4']	['D3', 'D4']	['D3', 'D4']
[]	[]	['D2']	['D1', 'D2']
[]	['D1']	['D1']	[]

[]
['D1', 'D2', 'D3', 'D4']
[]
Move 15

Q03: Set Operation

- Consider two input files FA and FB having elements (numbers) in sorted order. Two files can have common elements, as well a file can have multiples of same element (number. Implement the following operations and output the result in a different file using `O(1)` space. Also, output the total number of comparison operations done. Do not use any built-in function or utilities.
  - FA ∪ FB i.e. Union operation. The result file contains all unique elements of both files in sorted order.
  - FA ∩ FB i.e. Intersection operation.The result file contains all unique elements common to both files in sorted order.
  - FA - FB i.e. Difference operation. Result contains all those unique elements in FA which are not in FB.

Q04: Inplace MergeSort

- Given two sorted array A and B of elements (numbers), Merge these two array in place using O(1) space extra memory. After the merging operations, all elemnts of A will be less than or equal to first element of B and both A and B will still be in sorted order. Take two arrays as input from command line or using two files. Note: two arrays need not be of same size.
  - For example if the two arrays are
    - A=[6,10,15,20], and B=[3,4,5,19]
  - The output arrays will be
    - A=[3,4,5,6], and B=[10,15,19,20]

Q05: Circular Prime

- Given an input positive integer M, check if it is circular prime i.e. when digits of this number are rotated by any number of positions, it is still a prime number. The input number M should be taken as command line argument.
- For example, number 1193 is circular prime because all of its rotations 1931, 9311, 3119 are prime numbers
  -

Q06: Max Sum Subsequence

- Given an input sequence of numbers (both positive and negative), find the subsequence with maximum sum i.e. when values of the subsequence are added, the sum is maximum compared to any other subsequence. Note: The sequence of numbers will in a file, where filename is command line argument.
- For example, the sequence 2, -3, 1.5, -1, 3, -2, -3, 3
- The max sum subsequence is (1.5, -1, 3) = 3.5

Q07: Closest Dates

- Given N number of date of births (to be read from input file in the format YYYY-MM-DD), identify the pair of dates which are closest to each other. There can be more than one such pair. The input to the proram is a filename that contains all Date of Births.
  - For example, for following dates, the closest dates are
    - 2017-02-12 and 2017-02-07

```
2018-06-23
2017-02-12
2016-08-31
2016-05-15
2017-11-19
2016-09-30
2018-11-30
2017-09-15
2017-02-07
2018-04-01
```

Q08: Left Rotation of String

- Given a string `S` of size `n`, and positive integer `m`, rotate left the string characters by `m` using `O(1)` memory space.
- For example, if the string is "`abcdefghijkl`", and `m=5`, the output should be "`fghijklabcde`".
- Using memory space more than `O(1)` will make it infructous. Input string is taken as command line argument.

Q09: Finding Anagrams

- Given an input files of some number of words (one word per line), find all the anagrams in the file e.g. the words "`teacher`","`cheater`" and "`chartee`" are angrams of each other. All angrams of a word present in the file should be displayed on same line. You can use built in sort function of the programming language you are using.
- For example,
  - If file contains the following,

```
stop
pots
opts
ria
air
xyz
```

  - The output should be

```
stop pots opts
ria air
xyz
```

Q10: Find Duplicate Number

- Given input file containing `N+1` distinct numbers between `1` and `N` (both inclusive) in random order, find the duplicate number. Each number is on one line. You can use only `O(1)` memory space
- Note:
  - you can use any number of files and file size can be O(N)
  - Performing a sort requires O(N) memory space and thus not permitted

Q11

- Given two input positive integer M and N, identify all such positive integers between M and N (inclusive of both), such that the number is perfectly divisible by all of its digits. For example for number `1236`, it is divisible by `1`, `2`, `3`, and `6` and this number qualifies. The number `1234` does not qualify because it is perfectly divisible by `1`, `2`, and `3` but not by `4`.
  - Note:
    - Check if any of the digit is 0, then it is invalid input.
    - Digits can repeat e.g. 13131 qualifies

Q12

- Given input positive odd integer N, construct a box as shown below i.e. forward diagonals, horizonal and vertifical divisions. For example for N = 15, following should be output.