Design and Analysis of Algorithms

L I 0: Assignment-01

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

Resources

- ??
-

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 to server.
  – Program (java) should run on server.
  – Running on your laptop/lab-system has zero value.
- Assignment to be submitted to person's account with lowest USN value (lexicographic order) by due date.
- 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.

Assignment Logistics

- Make your group/team in the Google doc.
- There are total of 10 programming questions. All programming to be carried out in java.
- 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 should include
  - The java program
    - e.g. `Asn01P01.java`, `Asn01P02.java, ...`
  - README file (`Readme.txt`), should contain
    1. Team details (Names, USN)
    2. Contribution of each team member
    3. Instruction 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. (program output when ran the program)
    2. Total number of contributing (key) operations i.e. a computation of time complexity.

Q01: Compute N^2

- Given input as positive integer N, compute N^2 using addition, subtraction and comparison operation. The operation of multiplication, division, remainder, shift left/ right etc are not permitted. The computation of N^2 should be done in time complexity less than O(N), i.e. preferably O(log N). The simple solution of adding N times the number N itself would have time complexity of O(N).

Q02: Hanoi Towers with 2+K towers

- Implement Hanoi's tower using more than 2+K towers to transfer N discs from tower 1 to tower 2. The number K (greater than 1) of additional towers should be input parameter along with N. 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 (K=1) 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']	[]

Q03

- Given input N, using N-bit representation, the lowest value corresponds to all bits with value '0' and highest value corresponds to all bits with value '1'. Taking the lowest value as 1, and highest value as 2^N, Generate the N sets each having 2^N-1 numbers in it. The set S_1 corresponds to all those values where 1^{st} bit i.e. bit number 0 (Least Significant bit) is set to 0, and set S_N corresponds to all those values where N^{th} bit (Most Significant bit) is set to 0. For example, if N=3, three sets would be
  - S_1 (xx0) = 1,3,5,7
  - S_2 (x0x) = 1,5,2,6
  - S_3 (0xx) = 1,2,3,4

Q04

- Given two inputs positive integer M and N, find all consecutive prime numbers between M and N (inclusive of both M and N). For example, if M=20, and N=61, then program should output
  - 29,31
  - 41,43
  - 59,61

Q05

- Given two input positive integers M and N, compute all the perfect numbers between M and N (inclusive of both M and N). A perfect number is defined as the number which is equal to sum of all its factors other than itself. Example for first three perfect numbers are
  - `6 = 1+2+3`
  - `28 = 1+2+4+7+14`
  - `496 = 1+2+4+8+16+31+62+124+248`

Q06

- Construct 2-dimentional grid(array) of size MxN consisting of english letters (to be read from input file) and a set of english words (to be read from another file), identify all the english words that appear in the grid either horizontally, vertically or diagonally,.

Q07

- Given N number of date of births (to be read from input file in the format YYYY-MM-DD), identify the two dates which are closest to each other.

Q08

- **Grocery discount problem**. There was an advertisement in the paper from a grocery store providing prices of various items. The store made an offer that if a subscriber chooses grocery items totalling equal to a specific value N (neither less than nor greater than), then the subscriber will get 90% discount. A grocery item can be picked as many times as desired. The pricelist (item, price) is to be read from a file, the special offer price N is another positive integer. Assume prices of all grocery items are positive integers.

Q09

- Given input positive integer N, list all sequences (consisting of letters ‘H’ and ’T’) of length N, where two ’T’ does not appear together. For simplicity, consider H corresponds to Head and T corresponds to Tail when tossing a coin. For example, when N=3, the answer would be
  - HHH, HHT, HTH, THH, THT

Hint: Use recursion.

Q10

- Dyck words: Given input positive integer N, generate all possible dyck word i.e. a balanced string of left and right parentheses. For example for N=3, all possible dyck words of parentheses are

  ((( )))

  (( ) ( ))

  (( )) ( )

  ( ) (( ))

  ( ) ( ) ( )

  – Hint : Use Recursion

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`.

Q12

- Given input positive integer N, construct a box with lines depicting outline, diagonals, horizonal and vertifical divisions. The diagram will look slightly different from odd and even as below