BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Work Integrated Learning Programmes Division
M.Tech (Data Science and Engineering)

(S2-18_DSECFZG519)
(Data Structures and Algorithms Design)
Academic Year 2018-2019

Assignment 2 – PS2 - [INTEGER MULTIPLICATION] - [Weightage 12%]

1. Problem Statement

Multiplication of two n digit numbers using common grade-school algorithm takes n^2 multiplication operations. **Design a sub quadratic-time algorithm for multiplying two n digit integers.** For 2 integers A and B, use the following approach:

$A * B = (A1 * B1)10^n + (A1 * B2 + A2 * B1)\ 10^{n/2} + A2 * B2$

Requirements:

1. Identify which design strategy is employed here and give a brief explanation about the same.
2. Use recurrence relations to find the number of multiplication operations required using this approach. Briefly explain the steps involved.
3. ***"Multiplying big integers has applications to data security, where big integers are used in encryption schemes."*** - Substantiate the above statement with one example.
4. Implement the above problem statement using Python.

Input:

Input of two numbers A and B should be taken in as user input from **inputPS2.txt**.

223245
123456

Output:

Display intermediate values of A1, A2, B1 and B2. Also display the product of the two integers.

1st number, A: 223245
2nd number, B:123456

Intermediate Values of A1, B1 after partition:

A:223245 A1: 223 A2: 245

B:123456 B1: 123 B2: 456

Intermediate Values of A1, B1 after partition:

A:223 A1: 2 A2: 23

B:123 B1: 1 B2: 23

Intermediate Values of A1, B1 after partition:

A:23 A1: 2 A2: 3

B:23 B1: 2 B2: 3

Intermediate Values of A1, B1 after partition:

A:223 A1: 2 A2: 23

B:456 B1: 4 B2: 56

Intermediate Values of A1, B1 after partition:

A:23 A1: 2 A2: 3

B:56 B1: 5 B2: 6

Intermediate Values of A1, B1 after partition:

A:245 A1: 2 A2: 45

B:123 B1: 1 B2: 23

Intermediate Values of A1, B1 after partition:

A:45 A1: 4 A2: 5

B:23 B1: 2 B2: 3

Intermediate Values of A1, B1 after partition:

A:245 A1: 2 A2: 45

B:456 B1: 4 B2: 56

Intermediate Values of A1, B1 after partition:

A:45 A1: 4 A2: 5

B:56 B1: 5 B2: 6

Result:> 223245 x 123456 = 27560934720

Display the output in **outputPS2.txt**.

2. **Deliverables**

- Word document **designPS2_<group id>.docx** detailing your answers to requirements 1,2 and 3.
- **Zipped AS2_PS2_IM_[Group id].py package folder** containing all the modules classes and functions for the employee node, binary tree and the main body of the program.
- **inputPS2.txt** file used for testing
- **outputPS2.txt** file generated while testing

3. **Instructions**

- Do not use inbuilt data structures available in Python. The purpose of these assignments is for you to lean how these data structures and algorithms work.
- It is compulsory to use Python for implementation.
- Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full.
- For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.
- Make sure that your read, understand, and follow all the instructions
- Ensure that the input and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.

4. Deadline

- The strict deadline for submission of the assignment is **Sep 5th, 2019 EoD.**
- Late submissions won't be evaluated.

5. How to submit

- This is a group assignment.
- Each group has to make one submission (only one, no resubmission) of solutions.
- Each group should zip the deliverables and name the zipped file as below
- "ASSIGNMENT2_[BLR/HYD/DLH/PUN/CHE]_[B1/B2/…]_[G1/G2/…].zip"
- and upload in CANVAS in respective location under ASSIGNMENT Tab.
- Assignment submitted via means other than through CANVAS will not be graded.

6. Evaluation

- The assignment carries 12 Marks
- Grading will depend on
    - Efficiency of design (detailed in the design document)
        - Generic explanation copied off the internet will not be considered.
    - Every bug in the functionality will lead to negative marking.
    - Duplication of design document / code will be penalized.
    - Source code files which contain compilation errors will get at most 25% of the value of that question.
    - Fully executable code with all functionality.
- Late submissions will not be evaluated.

7. Readings

Text book: Algorithms Design: Foundations, Analysis and Internet Examples Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). Chapters: 5.2