Comp 251: Assignment 4

Instructor: Jérôme Waldispühl

Due on November 21 at 11h55:00

- Your solution must be submitted electronically on MyCourses.
- You are provided some starter code that you should fill in as requested. Add your code only where you are instructed to do so. You can add some helper methods, but this is at your own risk; helper methods could cause your methods to crash when called from other programs if not added responsibly. Do not modify the code in any other way and in particular, do not change the methods or constructors that are already given to you, do not import extra code and do not touch the method headers. The format that you see on the provided code is the only format accepted for programming questions. **Any failure to comply with these rules will give you an automatic 0.**
- The starter code includes a tester class. If your code fails those tests, it means that there is a mistake somewhere. Even if your code passes those tests, it may still contain some errors. We will grade your code with a more challenging set of examples. We therefore highly encourage you to modify that tester class, expand it and share it with other students on the myCourses discussion board. Do not include it in your submission.
- Your code should be properly commented and indented.
- **Do not change or alter the name of one of the files you must submit**. Files with the wrong name will not be graded. Make sure you are not changing file names by duplicating them. For example, main (2).java will not be graded. Make sure to double-check your zip file.
- Do not submit individual files. Include all your files into a .zip file and, when appropriate, answer the complementary quiz online on MyCourses.
- **You will automatically get 0 if the files you submitted on MyCourses do not compile.**
- To some extent, collaborations are allowed. These collaborations should not go as far as sharing code or giving away the answer. You must indicate on your assignments (i.e. as a comment at the beginning of your java source file) the names of the people with whom you collaborated or discussed your assignments (including members of the course staff). If you did not collaborate with anyone, you write "No collaborators". If asked, you should be able to orally explain your solution to a member of the course staff.
- It is your responsibility to guarantee that your assignment is submitted on time. We do not cover technical issues or unexpected difficulties you may encounter. Last minute submissions are at your own risk.
- Multiple submissions are allowed before the deadline. We will only grade the last submitted file. Therefore, we encourage you to submit as early as possible a preliminary version of your solution to avoid any last minute issue.

- Late submissions will receive a penalty of 20% per day. We will not accept any submission more than 72 hours after the deadline. The submission site will be closed, and there will be no exceptions, except medical.
- In exceptional circumstances, we can grant a small extension of the deadline (e.g. 24h) for medical reasons only. However, such request must be submitted before the deadline, and justified by a medical note from a doctor, which must also be submitted to the McGill administration.
- Violation of any of the rules above may result in penalties or even absence of grading. If anything is unclear, it is up to you to clarify it by asking either directly the course staff during office hours, by email at (`cs251@cs.mcgill.ca`) or on the discussion board on myCourses (recommended). Please, note that we reserve the right to make specific/targeted announcements affecting/extending these rules in class and/or on the website. It is your responsibility to monitor the course website and MyCourses for announcements.
- The course staff will answer questions about the assignment during office hours or in the online forum on MyCourses. We urge you to ask your questions as early as possible. We cannot guarantee that questions asked less than 24h before the submission deadline will be answered in time. In particular, we will not answer individual emails about the assignment that are sent sent the day of the deadline.
- You should compile all your files directly from command line without using a package, by using the command `javac *.java`

1. (50 points) **Part 1 : Programming exercise**
   . We want to compare the naive and Karatsuba divide-and-conquer methods to multiply two integers `x` and `y`. Download the java file `multiply.java` from the course web page. Here, your task is to implement a recursive version of these algorithms in the methods `naive(int size, int x, int y)` and `karatsuba(int size, int x, int y)`, and to use them to compare the efficiency of each algorithm (i.e. number of arithmetic operations). The variable `size` is the size of the integers `x` and `y`, and is defined as the number of bits used to encode them (Note: we assume that `x` and `y` have the same size).
   Each method (i.e. `naive` and `karatsuba`) will return an integer array `result` that stores the value of the multiplication in the first entry of the array (i.e. `result[0]`), and the cost of this computation in the second entry (i.e. `result[1]`). We define the cost as the number of brute force arithmetic operations of the (addition, subtraction, or multiplication) executed by the algorithm multiplied by the size (in bits) of the integers involved in this operation (Note: We ignore the multiplication by powers of 2 which can be executed using a bit shift. Of course, this is a crude approximation).
   In particular, for the base case (i.e. when the size of the integers is 1 bit), this cost will be 1 (brute force multiplication of two integers of size 1). In the induction case, the naive method executes 3 arithmetic operations of integer of size m (i.e. cost is $3 \cdot m$), in addition of the number of operations executed by each recursive call to the function. By contrast, the Karatsuba algorithm requires 6 arithmetic operations of size m on the top of the cost of the recursion.
   The output of your program will print a list of numbers such that, the first number of each row is the size of the integers that have been multiplied, the second number is the cost of the naive method, and the third number the cost of the Karatsuba method.

You will test your methods on integers of size 1 to 15. Complete the provided CSV file, `karatsuba.csv` with your results for both methods on integers of each size. Plot those results, and write a short report describing and explaining what you observe. Include the plot and the report in a file named `260xxxxxx.pdf`, editing the name to match your student ID.

Part 2 : write your computations with all the steps and justified results in a PDF file named 260xxxxxx.pdf. If this PDF is hand-written, make sure you scan it. Pictures will not be accepted.

Enter your results on the MyCourses quiz

2. (25 points) We remind the master method for determining the asymptotical behaviour of a recursive function.

   Theorem 1 (Master method) *Let $a \geq 1$ and $b \geq 1$ be two constants, and $f(n)$ a function. $\forall n \in \mathbb{N}^+$ we define $T(n)$ as:*

   $$T(n) = aT\Big(\frac{n}{b}\Big) + f(n), \textit{where } \tfrac{n}{b} \textit{ is interpreted as } \lfloor \tfrac{n}{b} \rfloor \textit{ or } \lceil \tfrac{n}{b} \rceil.$$

   Then, we can find asymptotical bounds for $T(n)$ such that:

   1. *If $f(n) = O(n^{\log_b a - \epsilon})$ with $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.*
   2. *If $f(n) = \Theta(n^{\log_b a} \cdot \log^p n)$, then $T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$.*
   3. *If $f(n) = \Omega(n^{\log_b a + \epsilon})$ with $\epsilon > 0$, and $a \cdot f\big(\frac{n}{b}\big) \leq cf(n), \forall n > n_0$ with $c < 1$ and $n_0 > 0$. Then $T(n) = \Theta(f(n))$.*

   When possible, apply the master theorem to find the asymptotic behaviour of $T(n)$ from the following recurrences. Show your work and justify your answer for each recurrence in the PDF file. Answer in the MyCourses quiz.

   (a) (5 points) $T(n) = 25 \cdot T(\frac{n}{5}) + n$

   (b) (5 points) $T(n) = 2 \cdot T(\frac{n}{3}) + n \cdot \log(n)$

   (c) (5 points) $T(n) = T(\frac{3n}{4}) + 1$

   (d) (5 points) $T(n) = 7 \cdot T(\frac{n}{3}) + n^3$

   (e) (5 points) $T(n) = T(n/2) + n(2 - \cos n)$

3. (25 points) Let T_A and T_B be two function returning the running time of algorithms A and B, defined by the recusions $T_A(n) = 7T_A(\frac{n}{2}) + n^2$ and $T_B(n) = \alpha T_B(\frac{n}{4}) + n^2$. Find the largest integer value of α for which algorithm B is asymptotically faster than A. Show your work and justify your answer in the PDF file. Answer in the MyCourses quiz.

You will submit multiply.java, karatsuba.csv and student_id.pdf in a single zip file in the MyCourses submission folder for Assignment 4. You will enter your solutions to question 2 and 3 on the MyCourses assignment 4 quiz. Your final answers must be entered in the quiz for you to receive a grade.