ASSIGNMENT 4: DIVIDE/DECREASE/TRANSFORM-AND-CONQUER ALGORITHM

due date: Sunday, April 3rd 2022 (23.59 WITA)

Aturan pengerjaan tugas:

1. Kerjakan **semua** soal yang ada secara singkat, padat, dan jelas.
2. Tugas boleh diketik/ditulis tangan (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah. Jika menggunakan tulis tangan, harap discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file. Tulis jawaban pada satu file pdf
3. Format penamaan tugas: **NamaLengkap_Kelas_NIM.extension**.
   Contoh: **GedeGanesha_6A_1610101001.pdf**.
   Pengumpulan tugas melalui e-learning Undiksha.
4. Anda diizinkan untuk berdiskusi dengan rekan Anda. Namun Anda harus menuliskan/menjelaskan jawaban Anda sendiri, dan paham dengan baik apa yang Anda tulis. Anda siap bertanggung jawab terhadap hasil pekerjaan Anda. Hasil pekerjaan yang memiliki kemiripan yang tinggi dengan pekerjaan mahasiswa lain mempengaruhi poin penilaian.
5. Tugas dinilai berdasarkan kerapian penulisan tugas teori, kerapian dan kejelasan program komputer, dan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Total nilai maksimum tugas ini adalah 100. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.

1 Polynomials multiplication

In this exercise, we investigate a divide-and-conquer approach to multiply two polynomials of the same order n (similar to the "Matrix multiplication" and "Large numbers multiplication" discussed in the lecture).

1. (*Naive polynomial multiplication algorithm*)
   Given two polynomials of the same order n as follows. Our goal is to compute $A(x)B(x)$.

$$A(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$
$$B(x) = b_0 + b_1x + b_2x^2 + \cdots + b_nx^n$$

   A naive way to perform the polynomials multiplication is by *direct multiplication*, as in the following example:

   Example

$$A(x) = 1 + 2x + 3x^2$$
$$B(x) = 3 + 2x + 2x^2$$

$$A(x)B(x) = (1 + 2x + 3x^2)(3 + 2x + 2x^2) = 3 + 8x + 15x^2 + 10x^3 + 6x^4 \qquad \square$$

   (a) Solve the following polynomials multiplication using naive algorithm!

$$A(x) = 2 + 5x + 3x^2 + x^3 - x^4$$
$$B(x) = 1 + 2x + 2x^2 + 3x^3 + 6x^4$$

   (b) Write a pseudocode for the naive algorithm to multiply two polynomials $A(x)$ and $B(x)$ that are of order n.

   (c) Compute the complexity of your naive algorithm. Represent it using an asymptotic notation!

2. (*Polynomials multiplication divide-and-conquer algorithm*)
   Given two polynomials of the same order n as follows. Our goal is to compute $A(x)B(x)$.

$$A(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$
$$B(x) = b_0 + b_1x + b_2x^2 + \cdots + b_nx^n$$

   How do we perform polynomials multiplication by Didive-and-Conquer? The algorithm is as follows.

   - Split $A(x)$ into $A_0(x)$ and $A_1(x)$, each contains $n/2$ terms:

$$A_0(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{\lceil n/2\rceil - 1}x^{\lceil n/2\rceil - 1}$$
$$A_1(x) = a_{\lceil n/2\rceil} + a_{\lceil n/2\rceil + 1}x + a_{\lceil n/2\rceil + 2}x^2 + \cdots + a_{n-\lceil n/2\rceil}x^{n-\lceil n/2\rceil}$$

     So that

$$A(x) = A_0(x) + A_1(x)x^{\lceil n/2\rceil}$$

   - Similarly, $B(x)$ can be split into $B_0(x)$ and $B_1(x)$, so that:

$$B(x) = B_0(x) + B_1(x)x^{\lceil n/2\rceil}$$

Hence:

$$A(x)B(x) = A_0(x)B_0(x) + (A_0(x)B_1(x) + A_1(x)B_0(x))\, x^{\lceil n/2 \rceil} + A_1(x)B_1(x)x^{2\lceil n/2 \rceil}$$

Task: Solve the following polynomials multiplication using the algorithm explained above. Write the steps clearly!

$$A(x) = 2 + 5x + 3x^2 + x^3 - x^4$$
$$B(x) = 1 + 2x + 2x^2 + 3x^3 + 6x^4$$

3. (*DnC-based polynomials multiplication: pseudocode and time complexity*)

   The divide-and-conquer algorithm explained in question 3 can be written in a pseudocode as follows:

Algorithm 1 Polynomials multiplication (divide-and-conquer, version 1)

procedure POLYMUL(A, B: polynomials, n: integer)
    declaration
        A_0, A_1, B_0, B_1: polynomials
        s: integer
    end declaration
    if $n = 0$ **then** **return** $A * B$ ▷ *scalar multiplication*
    else
        $s \leftarrow \lceil n/2 \rceil$
        $A_0 \leftarrow a_0 + a_1x + a_2x^2 + \cdots + a_{s-1}x^{s-1}$
        $A_1 \leftarrow a_sx^s + a_{s+1}x^{s+1} + a_{s+2}x^{s+2} + \cdots + a_nx^{n-s}$
        $B_0 \leftarrow b_0 + b_1x + b_2x^2 + \cdots + b_{s-1}x^{s-1}$
        $B_1 \leftarrow b_sx^s + b_{s+1}x^{s+1} + b_{s+2}x^{s+2} + \cdots + b_nx^{n-s}$
        return POLYMUL(A_0, B_0, s) + POLYMUL(A_0, B_1, s) + POLYMUL(A_1, B_0, s) $*x^s$ + POLYMUL(A_1, B_1, s) $*x^{2s}$
    end if
end procedure

   (a) Check if the pseudocode really matches the mathematical computation explained in question 3.

   (b) Write the time complexity function of the divide-and-conquer algorithm in question 3 in a recursive formula. Using Master Theorem, compute the asymptotic time complexity!

4. (*DnC-based polynomials multiplication: improvement*)

   Now we want to modify the divide-and-conquer algorithm for the polynomials multiplication given in question 2. We will reduce the number of multiplications performed in the algorithm.

   In question 2, we have:

$$A(x)B(x) = A_0(x)B_0(x) + (A_0(x)B_1(x) + A_1(x)B_0(x))\, x^{\lceil n/2 \rceil} + A_1(x)B_1(x)x^{2\lceil n/2 \rceil}$$

   There are 4 multiplications and 3 additions of polynomials of order n. We will reduce the number of multiplications to 3, but with a consequence that the number of additions is increased.

   Define:

$$Y(x) = (A_0(x) + A_1(x)) \times (B_0(x) + B_1(x))$$
$$U(x) = A_0(x)B_0(x)$$
$$Z(x) = A_1(x)B_1(x)$$

Then:

$$Y(x) - U(x) - Z(x) = A_0(x)B_1(x) + A_1(x)B_0(x)$$

so that:

$$\begin{aligned} A(x)B(x) &= A_0(x)B_0(x) + (A_0(x)B_1(x) + A_1(x)B_0(x))x^{\lceil n/2 \rceil} + A_1(x)B_1(x)x^{2\lceil n/2 \rceil} \\ &= U(x) + (Y(x) - U(x) - Z(x))\, x^{\lceil n/2 \rceil} + Z(x)x^{2\lceil n/2 \rceil} \end{aligned}$$

Note that in this algorithm, there are only three multiplications, namely $U(x)$, $Y(x)$, and $Z(x)$.

Task: Solve the following polynomials multiplication using the algorithm explained above. Write the steps clearly!

$$\begin{aligned} A(x) &= 2 + 5x + 3x^2 + x^3 - x^4 \\ B(x) &= 1 + 2x + 2x^2 + 3x^3 + 6x^4 \end{aligned}$$

5. (*DnC poly-multiplication improvement: pseudocode*)

   The divide-and-conquer algorithm explained in question 3 can be written in a pseudocode as follows:

Algorithm 2 Polynomials multiplication (divide-and-conquer, version 2)

procedure POLYMUL2(A, B: polynomials, n: integer)
    declaration
        $A_0, A_1, B_0, B_1, U, Y, Z$: polynomials
        s: integer
    end declaration
    if $n = 0$ **then**
        return $A * B$ ▷ *scalar multiplication*
    else
        $s \leftarrow \lceil n/2 \rceil$
        $A_0 \leftarrow a_0 + a_1x + a_2x^2 + \cdots + a_{s-1}x^{s-1}$
        $A_1 \leftarrow a_sx^s + a_{s+1}x^{s+1} + a_{s+2}x^{s+2} + \cdots + a_nx^{n-s}$
        $B_0 \leftarrow b_0 + b_1x + b_2x^2 + \cdots + b_{s-1}x^{s-1}$
        $B_1 \leftarrow b_sx^s + b_{s+1}x^{s+1} + b_{s+2}x^{s+2} + \cdots + b_nx^{n-s}$
        $Y \leftarrow$ POLYMUL2($A_0 + A_1, B_0 + B_1, s$)
        $U \leftarrow$ POLYMUL2(A_0, B_0, s)
        $Z \leftarrow$ POLYMUL2(A_1, B_1, s)
        return $U + (Y - U - Z) * x^s + Z * z^{2s}$
    end if
end procedure

Task: Solve the following polynomials multiplication by algorithm POLYMUL2. Write the steps clearly!

$$\begin{aligned} A(x) &= 2 + 5x + 3x^2 + x^3 - x^4 \\ B(x) &= 1 + 2x + 2x^2 + 3x^3 + 6x^4 \end{aligned}$$

6. After computing the result of $A(x)B(x)$, check if the three different algorithms above give the same result. If not, explain why!

2 Fun with "conquer"

1. (*LU-decomposition vs Gaussian elimination*)

   (a) Solve the following system by Gaussian elimination

$$\begin{aligned} 2x_1 + x_2 - x_3 &= 4 \\ x_1 + x_2 + x_3 &= 2 \\ x_1 - x_2 + 2x_3 &= 8 \end{aligned}$$

   (b) Which method do you think more efficient in general for matrix operation (such as: solving linear system, inverse, etc.): LU decomposition or the Gaussian elimination? Explain your answer!

2. (*Presorting*)

   Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ be two sets of numbers. Consider the problem of finding their intersection, i.e., the set C of all the numbers that are in both A and B.

   (i) Design a brute-force algorithm for solving this problem and determine its efficiency class.

   (ii) Design a presorting-based algorithm for solving this problem and determine its efficiency class.

3. (*Maximizing product two numbers*)

   Consider the problem of finding, for a given positive integer n, the pair of integers whose sum is n and whose product is as large as possible. Design an efficient algorithm for this problem and indicate its efficiency class.

4. (*Transformation of the edge coloring problem*)

   The *graph-coloring problem* is usually stated as the **vertex-coloring problem**: Given a graph G, assign the smallest number of colors to vertices of G, so that no two adjacent vertices have the same color.

   Consider the **edge-coloring problem**: Assign the smallest number of colors possible to edges of a given graph so that no two edges with the same endpoint are the same color.

   Explain how the edge-coloring problem can be reduced to a vertex-coloring problem. (*Hint:* create a new graph)

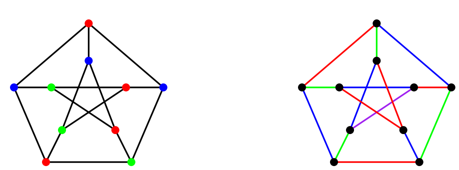

Figure 1: Example of vertex coloring (left) and edge coloring (right)