Laporan Tugas Kecil 2 - Strategi Algoritma (IF2211)

Membuat Pustaka untuk Perkalian Polinom dengan Algoritma *Divide and Conquer*

Oleh:

Muhammad Hasan – 13518012 – K03

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER 2 TAHUN 2019/2020

Daftar Isi

BAB 1 - Algoritma Bruteforce dan *Divide and Conquer* untuk Perkalian Polinom

1.1 Algoritma Bruteforce untuk Perkalian Polinom

Algoritma *bruteforce* dalam mengalikan polinom cukup *straightforward*. Jika kita memiliki dua buah polinom $A(x)$ dan $B(x)$ berderajat n, dengan:

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$
$$B(x) = b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_0$$

Maka hasil perkaliannya adalah dengan mengalikan setiap suku dari polinomial A dengan polinomial B, seperti uraian berikut:

$$\begin{aligned} A(x)B(x) &= (a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0)B(x) \\ &= a_n x^n B(x) + a_{n-1} x^{n-1} B(x) + \cdots + a_0 B(x) \\ &= (a_n b_n x^{2n} + \cdots + a_n a_0 x^n) + (a_{n-1} b_n x^{2n-1} + \cdots + a_{n-1} b_0 x^{n-1}) + \cdots + (a_0 b_n x^n + \cdots + a_0 b_0) \\ &= a_n b_n x^{2n} + (a_n b_{n-1} + a_{n-1} b_n) x^{2n-1} + \cdots + (a_1 b_0 + a_0 b_1) x + a_0 b_0 \end{aligned}$$

Hasil perkalian itu dapat diformulasikan sebagai berikut:

$$A(x)B(x) = \sum_{k=0}^{2n} c_k x^k$$

Dengan $c_k = \sum_{i=0}^{k} a_i b_{k-i}$.

Tentunya, cara *bruteforce* ini dapat dilakukan pada perkalian polinomial yang berbeda derajat. Kemudian, dapat dilihat bahwa pada cara ini, setiap elemen dikalikan pada A dikalikan n kali dengan elemen-elemen pada B, baru kemudian ditambahkan semua. Oleh karena itu, pada algoritma *bruteforce* ini, kita akan mempunyai hasil kompleksitas waktu $O(n^2)$.

1.2 Algoritma Divide and Conquer untuk Perkalian Polinom

Untuk dua polinomial yang berderajat sama, kita dapat mengalikannya dengan Algoritma *Divide and Conquer*. Jika kita memiliki dua buah polinom $A(x)$ dan $B(x)$ berderajat n, dengan:

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$
$$B(x) = b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_0$$

Kedua polinomal tersebut dapat kita bagi (*divide*), contohnya untuk polinomal A, kita bisa bagi menjadi:

$$A_0(x) = a_0 + a_1 x + \cdots + a_{\lfloor \frac{n}{2} \rfloor} x^{\lfloor \frac{n}{2} \rfloor - 1}$$

$$A_1(x) = a_{\lfloor \frac{n}{2} \rfloor} + a_{\lfloor \frac{n}{2} \rfloor + 1} x + \cdots + a_n x^{n - \lfloor \frac{n}{2} \rfloor}$$

sehingga

$$A(x) = A_0(x) + A_1(x) x^{\lfloor \frac{n}{2} \rfloor}$$

Dengan cara yang sama pada polinomal B, kita bisa dapat juga:

$$B_0(x) = b_0 + b_1 x + \cdots + b_{\lfloor \frac{n}{2} \rfloor} x^{\lfloor \frac{n}{2} \rfloor - 1}$$

$$B_1(x) = b_{\lfloor \frac{n}{2} \rfloor} + b_{\lfloor \frac{n}{2} \rfloor + 1} x + \cdots + b_n x^{n - \lfloor \frac{n}{2} \rfloor}$$

$$B(x) = B_0(x) + B_1(x) x^{\lfloor \frac{n}{2} \rfloor}$$

Dengan cara tersebut, kita bisa dapatkan perkalian polinom A dan polinom B sebagai berikut:

$$\begin{aligned} A(x)B(x) &= \left(A_0(x) + A_1(x) x^{\lfloor \frac{n}{2} \rfloor}\right)\left(B_0(x) + B_1(x) x^{\lfloor \frac{n}{2} \rfloor}\right) \\ &= A_0(x)B_0(x) + A_0(x)B_1(x) x^{\lfloor \frac{n}{2} \rfloor} + A_1(x)B_0(x) x^{\lfloor \frac{n}{2} \rfloor} + A_1(x)B_1(x) x^{2\lfloor \frac{n}{2} \rfloor} \end{aligned}$$

Perkalian tersebut dapat kita modifikasi lagi, misalkan:

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

Dengan ini kita bisa lakukan tahap *conquer*, sehingga hasil perkalian menjadi:

$$A(x)B(x) = U(x) + \big(Y(x) - U(x) - Z(x)\big)x^{\left\lfloor \frac{n}{2} \right\rfloor} + Z(x)x^{2\left\lfloor \frac{n}{2} \right\rfloor}$$

Kompleksitas algoritma ini dapat kita hitung sebagai berikut:

Kita akan punya jumlah operasi $T(n)$ didefinisikan sebagai:

$$T(n) = \begin{cases} 1 \text{ (untuk } n = 1) \\ 3T\left(\frac{n}{2}\right) + cn \end{cases}$$

Dengan c suatu konstanta

Misalkan $n = 2^h$, kita akan punya

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{2}\right) + cn \\ &= 3\left(3T\left(\frac{n}{2^2}\right) + \frac{cn}{2}\right) + cn \\ &= 3^2T\left(\frac{n}{2^2}\right) + \left(1 + \frac{3}{2}\right)cn \\ &= 3^3T\left(\frac{n}{2^3}\right) + \left(1 + \frac{3}{2} + \left(\frac{3}{2}\right)^2\right)cn \\ &\vdots \\ &= 3^hT\left(\frac{n}{2^h}\right) + \left(\sum_{i=0}^{h-1}\left(\frac{3}{2}\right)^i\right)cn \end{aligned}$$

Maka diperoleh $3^h = \left(2^{\log 3}\right)^h = 2^{h \log 3} = (2^h)^{\log 3} = n^{\log 3} = n^{1.585}$, dan

$$\sum_{i=0}^{h-1} \left(\frac{3}{2}\right)^i = \frac{\left(\frac{3}{2}\right)^h - 1}{\frac{3}{2} - 1} = 2\left(\frac{3}{2}\right)^h - 2 = 2n^{\log 3 - 1} - 2$$

Sehingga didapat kompleksitas waktu:

$$T(n) = O\left(n^{\log 3} T(1) + 2cn^{\log 3}\right) = O\left(n^{\log 3}\right) \approx O(n^{1.585})$$

BAB 2 – *Source Code* Program

2.1 *Checklist* Tabel Program

Poin	Ya	Tidak
Program berhasil dikompilasi	✓	
Program berhasil *running*	✓	
Program dapat menerima input dan menuliskan output	✓	
Luaran sudah benar untuk semua data uji	✓	

2.2 Source Code Program

Program dibuat dengan menggunakan bahasa pemrograman C++

Berikut adalah *source code* program **polinom.hpp**:

```
#ifndef _POLINOM_HPP_
#define _POLINOM_HPP_

#include <bits/stdc++.h>

using namespace std;

int jumlah_operasi = 0;

class Polinom {
  protected:
  int derajat;
  deque<int> polinom;

  public:
  Polinom() {
    derajat = -1;
```

```
    polinom.clear();
  }

  Polinom(int n) {
    derajat = n;
    polinom.resize(n + 1);
    for (int i = 0; i <= derajat; i++) {
      polinom[i] = rand() % 200 - 100;
    }
  }

  Polinom(const Polinom& p) {
    polinom = p.polinom;
    derajat = p.derajat;
  }

  Polinom(const Polinom& p, int s, int e) {
    derajat = e - s;
    polinom.resize(e - s + 1);
    for (int i = s, j = 0; i <= e; i++, j++) {
      polinom[j] = p.polinom[i];
    }
  }

  Polinom(const deque<int>& p) {
    polinom = p;
    derajat = (int) polinom.size() - 1;
  }

  Polinom& operator=(const Polinom& p) {
    polinom = p.polinom;
    derajat = p.derajat;
    return *this;
  }

  void add(int x) {
    derajat++;
    polinom.push_back(x);
  }

  void cetak_polinom() {
    if (derajat == 0) {
      cout << polinom[0] << endl;
      return;
    }
```

```
    for (int i = 0; i <= derajat; i++) {
      if (polinom[i] == 0) continue;
      if (i == 0) {
        cout << polinom[i] << " ";
        continue;
      }
      if (polinom[i] > 0) {
        if (i > 0) cout << "+ ";
        cout << polinom[i] << "x^" << i << " ";
      } else if (i == 0) {
        cout << polinom[i] << " ";
      } else {
        cout << "- " << -polinom[i] << "x^" << i << " ";
      }
    }
    cout << endl << endl;
  }

  friend Polinom operator+(const Polinom& p1, const Polinom& p2) {
    Polinom result;
    result.derajat = max(p1.derajat, p2.derajat);
    result.polinom.resize(result.derajat + 1);
    for (int i = 0; i <= result.derajat; i++) {
      int c_1 = (i <= p1.derajat ? p1.polinom[i] : 0);
      int c_2 = (i <= p2.derajat ? p2.polinom[i] : 0);
      result.polinom[i] = c_1 + c_2;
    }
    while (result.derajat >= 0 && result.polinom.back() == 0) {
      result.derajat--;
      result.polinom.pop_back();
    }
    return result;
  }

  friend Polinom operator-(const Polinom& p1, const Polinom& p2) {
   Polinom result;
    result.derajat = max(p1.derajat, p2.derajat);
    result.polinom.resize(result.derajat + 1);
    for (int i = 0; i <= result.derajat; i++) {
      int c_1 = (i <= p1.derajat ? p1.polinom[i] : 0);
      int c_2 = (i <= p2.derajat ? p2.polinom[i] : 0);
      result.polinom[i] = c_1 - c_2;
    }
    while (result.derajat >= 0 && result.polinom.back() == 0) {
```

```
      result.derajat--;
      result.polinom.pop_back();
    }
    return result;
  }

  friend Polinom kali_x_pangkat(Polinom &p, int n) {
    p.derajat += n;
    for (int i = 0; i < n; i++) p.polinom.push_front(0);
    jumlah_operasi += n + 1;
    return p;
  }

  friend Polinom hasil_kali_bf(Polinom p1, Polinom p2) {
    Polinom hasil_polinom;
    deque<int> hasil;
    int derajat = p1.derajat;
    int hasil_derajat = p1.derajat + p2.derajat;
    jumlah_operasi += 2;

    for (int i = 0; i <= p1.derajat + p2.derajat; i++) {
      hasil.push_back(0);
      jumlah_operasi++;
    }

    for (int i = 0; i <= p1.derajat; i++) {
      for (int j = 0; j <= p2.derajat; j++) {
        hasil[i + j] += (p1.polinom[i] * p2.polinom[j]);
        jumlah_operasi += 2;
      }
    }

    hasil_polinom.derajat = p1.derajat + p2.derajat;
    jumlah_operasi++;
    for (int i = 0; i <= p1.derajat + p2.derajat; i++) {
      hasil_polinom.polinom.push_back(hasil[i]);
      jumlah_operasi++;
    }

    return hasil_polinom;
  }

  friend Polinom hasil_kali_dnc(const Polinom &p1, const Polinom &p2) {
    if (p1.derajat == 0) {
      jumlah_operasi++;
```

```
      deque<int> dq(1, p1.polinom[0] * p2.polinom[0]);
      return Polinom(dq);
    }
    int len = p1.derajat;
    int mid = (len + 1) / 2;
    int m = len / 2 - (len % 2 == 0);
    jumlah_operasi += 4;

    Polinom A0 = Polinom(p1, 0, m);
    Polinom A1 = Polinom(p1, m + 1, len);
    Polinom B0 = Polinom(p2, 0, m);
    Polinom B1 = Polinom(p2, m + 1, len);

    Polinom Y = hasil_kali_dnc(A0 + A1, B0 + B1);
    Polinom U = hasil_kali_dnc(A0, B0);
    Polinom Z = hasil_kali_dnc(A1, B1);
    Polinom T = Y - U - Z;

    return U + kali_x_pangkat(T, mid) + kali_x_pangkat(Z, 2 * mid);
  }
};

#endif
```

Berikut adalah *source code* program **main.cpp**:

```
#include <bits/stdc++.h>
#include "polinom.hpp"

using namespace std;

int main() {
  srand(time(0));
  clock_t timer;
  int derajat;

  cout << "Masukkan derajat polinom yang diinginkan : ";
  cin >> derajat;

  Polinom A(derajat), B(derajat);
  cout << "Polinom A(x) yang terbuat adalah:" << endl;
  cout << "A(x) = ";
```

```
  A.cetak_polinom();

  cout << "Polinom B(X) yang terbuat adalah:" << endl;
  cout << "B(x) = ";
  B.cetak_polinom();

  timer = clock();
  int operasi_sekarang = jumlah_operasi;

  Polinom C = hasil_kali_bf(A, B);
  cout << "Hasil kali bruteforce adalah:" << endl;
  cout << "A(x)B(x) = ";
  C.cetak_polinom();

  int operasi_setelah = jumlah_operasi;
  cout << "Terjadi sebanyak " << operasi_setelah - operasi_sekarang << " jumlah
 operasi kali dan tambah" << endl;
  cout << "Waktu penghitungan : " << (1000.0) * (clock() - timer) / CLOCKS_PER_
SEC << " ms" << endl << endl;

  timer = clock();
  operasi_sekarang = jumlah_operasi;

  Polinom D = hasil_kali_dnc(A, B);
  cout << "Hasil kali Divide and Conquer adalah:\n";
  cout << "A(x)B(x) = ";
  D.cetak_polinom();

  operasi_setelah = jumlah_operasi;
  cout << "Terjadi sebanyak " << operasi_setelah - operasi_sekarang << " jumlah
 operasi kali dan tambah" << endl;
  cout << "Waktu penghitungan : " << (1000.0) * (clock() - timer) / CLOCKS_PER_
SEC << " ms" << endl;

  return 0;
}
```

BAB 3 – Screenshot Input-Output Program

3.1 Spesifikasi Personal Computer

Spesifikasi *Personal Computer* (PC) yang digunakan dalam menjalankan program adalah sebagai berikut:

Processor	RAM	HardDrive
Intel® Core™ i7-8550U CPU @ 1.80 GHz – 1.99 GHz	DDR3 – 16GB	SSD SATA 3 – 512 GB

3.2 Screenshot Input-Output Program

Terdapat 4 uji *test case* sebagai berikut:

3.2.1 Test Derajat 5

```
Masukkan derajat polinom yang diinginkan : 5
Polinom A(x) yang terbuat adalah:
A(x) = -99 + 10x^1 + 9x^2 + 9x^3 + 62x^4 + 63x^5

Polinom B(X) yang terbuat adalah:
B(x) = 19 + 27x^1 - 52x^2 + 16x^3 - 25x^4 + 52x^5

Hasil kali bruteforce adalah:
A(x)B(x) = -1881 - 2483x^1 + 5589x^2 - 1690x^3 + 3588x^4 - 2851x^5 - 1084x^6 - 2041x^7 - 74x^8 + 1649x^9 + 3276x^10

Terjadi sebanyak 97 jumlah operasi kali dan tambah
Waktu penghitungan : 6 ms

Hasil kali Divide and Conquer adalah:
A(x)B(x) = -1881 - 2483x^1 + 5589x^2 - 1690x^3 + 3588x^4 - 2851x^5 - 1084x^6 - 2041x^7 - 74x^8 + 1649x^9 + 3276x^10

Terjadi sebanyak 117 jumlah operasi kali dan tambah
Waktu penghitungan : 3 ms
```

3.2.1 Test Derajat 10

```
Masukkan derajat polinom yang diinginkan : 10
Polinom A(x) yang terbuat adalah:
A(x) = 35 - 89x^1 - 24x^2 - 36x^3 - 25x^4 + 19x^5 - 43x^6 + 81x^7 - 36x^8 - 79x^9 - 89x^10

Polinom B(X) yang terbuat adalah:
B(x) = 58 - 97x^1 + 67x^2 + 70x^3 - 38x^4 - 45x^5 - 35x^6 - 16x^7 + 77x^8 + 19x^9 - 30x^10

Hasil kali bruteforce adalah:
A(x)B(x) = 2030 - 8557x^1 + 9586x^2 - 3273x^3 - 7126x^4 + 1242x^5 - 4840x^6 + 13395x^7 - 3967x^8 - 2814x^9 + 3
400x^10 - 1146x^11 - 14458x^12 - 1687x^13 + 4701x^14 + 12196x^15 + 4436x^16 - 7773x^17 - 7274x^18 + 679x^19 +
2670x^20

Terjadi sebanyak 287 jumlah operasi kali dan tambah
Waktu penghitungan : 32 ms

Hasil kali Divide and Conquer adalah:
A(x)B(x) = 2030 - 8557x^1 + 9586x^2 - 3273x^3 - 7126x^4 + 1242x^5 - 4840x^6 + 13395x^7 - 3967x^8 - 2814x^9 + 3
400x^10 - 1146x^11 - 14458x^12 - 1687x^13 + 4701x^14 + 12196x^15 + 4436x^16 - 7773x^17 - 7274x^18 + 679x^19 +
2670x^20

Terjadi sebanyak 347 jumlah operasi kali dan tambah
Waktu penghitungan : 12 ms
```

3.2.2 Test Derajat 20

```
Masukkan derajat polinom yang diinginkan : 20
Polinom A(x) yang terbuat adalah:
A(x) = -75 + 47x^1 + 23x^2 + 79x^3 + 69x^4 + 50x^5 + 53x^6 + 97x^7 - 65x^8 - 1x^9 - 53x^10 - 22x^11 - 17x^12 -
 70x^13 - 75x^14 - 44x^15 + 92x^16 + 10x^17 + 78x^18 + 20x^19 - 9x^20

Polinom B(X) yang terbuat adalah:
B(x) = -83 + 14x^1 - 66x^2 - 18x^3 + 10x^4 + 98x^5 + 7x^6 + 40x^7 - 53x^8 + 68x^9 - 20x^10 + 34x^11 + 2x^12 -
43x^13 - 15x^14 + 81x^15 - 51x^16 + 32x^17 - 11x^18 - 80x^19 + 13x^20

Hasil kali bruteforce adalah:
A(x)B(x) = 6225 - 4951x^1 + 3699x^2 - 7987x^3 - 7735x^4 - 15692x^5 - 5364x^6 - 11478x^7 + 16803x^8 - 7039x^9 +
 19479x^10 + 5481x^11 + 18549x^12 + 8960x^13 + 10607x^14 - 8295x^15 + 9930x^16 - 11419x^17 - 8408x^18 - 11528x
^19 - 17612x^20 + 7341x^21 + 571x^22 - 1499x^23 - 1573x^24 - 3624x^25 - 8405x^26 + 15520x^27 - 1871x^28 - 216x
^29 - 1515x^30 + 5370x^31 - 1134x^32 + 14413x^33 - 370x^34 - 7295x^35 + 637x^36 - 6618x^37 - 487x^38 + 980x^39
 - 117x^40

Terjadi sebanyak 967 jumlah operasi kali dan tambah
Waktu penghitungan : 20 ms

Hasil kali Divide and Conquer adalah:
A(x)B(x) = 6225 - 4951x^1 + 3699x^2 - 7987x^3 - 7735x^4 - 15692x^5 - 5364x^6 - 11478x^7 + 16803x^8 - 7039x^9 +
 19479x^10 + 5481x^11 + 18549x^12 + 8960x^13 + 10607x^14 - 8295x^15 + 9930x^16 - 11419x^17 - 8408x^18 - 11528x
^19 - 17612x^20 + 7341x^21 + 571x^22 - 1499x^23 - 1573x^24 - 3624x^25 - 8405x^26 + 15520x^27 - 1871x^28 - 216x
^29 - 1515x^30 + 5370x^31 - 1134x^32 + 14413x^33 - 370x^34 - 7295x^35 + 637x^36 - 6618x^37 - 487x^38 + 980x^39
 - 117x^40

Terjadi sebanyak 1027 jumlah operasi kali dan tambah
Waktu penghitungan : 17 ms
```

3.2.3 Test Derajat 50

```
Masukkan derajat polinom yang diinginkan : 50
Polinom A(x) yang terbuat adalah:
A(x) = -92 - 34x^1 + 9x^2 - 63x^3 + 29x^4 - 62x^5 - 89x^6 + 10x^7 + 80x^8 - 44x^9 + 64x^10 + 45x^11 - 24x^12 + 20x^13 - 98x^14 + 49x^15 - 60x^16 + 14x^17 + 94x^18 + 82x^19
 - 100x^20 + 82x^21 + 44x^22 - 64x^23 - 51x^24 + 78x^25 - 71x^26 - 58x^27 - 10x^28 + 31x^29 + 81x^30 - 10x^31 - 92x^32 + 80x^33 + 34x^34 - 46x^35 - 69x^36 + 33x^37 - 87x^38
 + 14x^39 + 46x^40 - 28x^41 - 49x^42 - 19x^43 - 64x^44 + 52x^45 + 84x^46 + 74x^47 - 68x^48 + 79x^49 + 67x^50

Polinom B(X) yang terbuat adalah:
B(x) = 80 + 27x^1 - 58x^2 - 57x^3 - 61x^4 - 12x^5 + 42x^6 + 44x^7 + 33x^8 + 48x^9 - 47x^10 - 31x^11 + 91x^12 - 18x^13 + 97x^14 + 60x^15 + 96x^16 + 84x^17 + 12x^18 + 22x^19
 + 28x^20 + 80x^21 - 19x^22 + 49x^23 + 20x^24 + 30x^25 + 50x^26 - 41x^27 - 42x^28 + 27x^29 - 69x^30 - 27x^31 - 1x^32 + 25x^33 + 82x^34 + 8x^35 + 68x^36 + 51x^37 + 19x^38 - 4
2x^39 + 69x^40 - 76x^41 - 50x^42 + 9x^43 + 71x^44 + 11x^45 - 94x^46 - 49x^47 + 58x^48 - 46x^49 + 36x^50

Hasil kali bruteforce adalah:
A(x)B(x) = -7360 - 5204x^1 + 5138x^2 + 2419x^3 + 7647x^4 + 2142x^5 - 10890x^6 - 1401x^7 + 10199x^8 - 1221x^9 + 6330x^10 + 5253x^11 - 23183x^12 - 6488x^13 - 20865x^14 - 1699
9x^15 + 5742x^16 - 12032x^17 + 179x^18 + 4336x^19 - 23453x^20 - 35433x^21 - 2481x^22 - 22506x^23 - 239x^24 + 18059x^25 - 17961x^26 + 17158x^27 - 4856x^28 - 5388x^29 + 18269
x^30 + 16635x^31 - 27684x^32 + 23725x^33 + 15017x^34 - 12359x^35 + 7719x^36 - 3108x^37 - 14525x^38 + 1407x^39 - 18774x^40 - 7313x^41 + 24884x^42 - 32147x^43 + 5317x^44 + 15
172x^45 + 27263x^46 + 23652x^47 - 12214x^48 - 31553x^49 - 25050x^50 + 2542x^51 - 7818x^52 - 8345x^53 + 8007x^54 + 30169x^55 - 10237x^56 - 24001x^57 + 7860x^58 + 5781x^59 -
42619x^60 + 11238x^61 + 20421x^62 + 20396x^63 - 24517x^64 + 18542x^65 + 38011x^66 + 3544x^67 - 6963x^68 + 23139x^69 + 13771x^70 - 7131x^71 - 10765x^72 + 29183x^73 + 810x^74
 - 7165x^75 - 24573x^76 - 2579x^77 + 3224x^78 - 6070x^79 - 10334x^80 + 10964x^81 - 651x^82 + 23792x^83 + 13233x^84 + 14176x^85 - 8787x^86 + 7812x^87 - 15976x^88 + 8800x^89
+ 16270x^90 - 5084x^91 - 21699x^92 - 332x^93 + 8568x^94 - 1057x^95 - 14493x^96 + 7091x^97 - 2196x^98 - 238x^99 + 2412x^100

Terjadi sebanyak 5407 jumlah operasi kali dan tambah
Waktu penghitungan : 54 ms

Hasil kali Divide and Conquer adalah:
A(x)B(x) = -7360 - 5204x^1 + 5138x^2 + 2419x^3 + 7647x^4 + 2142x^5 - 10890x^6 - 1401x^7 + 10199x^8 - 1221x^9 + 6330x^10 + 5253x^11 - 23183x^12 - 6488x^13 - 20865x^14 - 1699
9x^15 + 5742x^16 - 12032x^17 + 179x^18 + 4336x^19 - 23453x^20 - 35433x^21 - 2481x^22 - 22506x^23 - 239x^24 + 18059x^25 - 17961x^26 + 17158x^27 - 4856x^28 - 5388x^29 + 18269
x^30 + 16635x^31 - 27684x^32 + 23725x^33 + 15017x^34 - 12359x^35 + 7719x^36 - 3108x^37 - 14525x^38 + 1407x^39 - 18774x^40 - 7313x^41 + 24884x^42 - 32147x^43 + 5317x^44 + 15
172x^45 + 27263x^46 + 23652x^47 - 12214x^48 - 31553x^49 - 25050x^50 + 2542x^51 - 7818x^52 - 8345x^53 + 8007x^54 + 30169x^55 - 10237x^56 - 24001x^57 + 7860x^58 + 5781x^59 -
42619x^60 + 11238x^61 + 20421x^62 + 20396x^63 - 24517x^64 + 18542x^65 + 38011x^66 + 3544x^67 - 6963x^68 + 23139x^69 + 13771x^70 - 7131x^71 - 10765x^72 + 29183x^73 + 810x^74
 - 7165x^75 - 24573x^76 - 2579x^77 + 3224x^78 - 6070x^79 - 10334x^80 + 10964x^81 - 651x^82 + 23792x^83 + 13233x^84 + 14176x^85 - 8787x^86 + 7812x^87 - 15976x^88 + 8800x^89
+ 16270x^90 - 5084x^91 - 21699x^92 - 332x^93 + 8568x^94 - 1057x^95 - 14493x^96 + 7091x^97 - 2196x^98 - 238x^99 + 2412x^100

Terjadi sebanyak 4036 jumlah operasi kali dan tambah
Waktu penghitungan : 50 ms
```