Multiplication de 2 polynomes de degre n

$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$
$B(x) = b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_0$
$C(x) = A(x) \times B(x) = c_{2n} x^{2n} + c_{2n-1} x^{2n-1} + \cdots + c_0$

Donc:

$$c_i = \sum_j a_j b_{i-j} \tag{1}$$

Il y a donc $(n+1)^2$ multiplications et n^2 addition

Ca c'est l'algo naif de multiplication de polynome.

Karatsuba algorithm

Voir le lien wikipedia. (Deja implem en TP de sup TinyBistro).

$(P_1 \times P_2)(x) = Q_1(x) \times Q_2(x) \times x^n + (Q_1(x) \times R_2(x)) + Q_2(x) \times R_1(x))x^{n/2}$

```
def Karatsuba(P1, P2, n):
  if n == 1:
    return [P1[0] + P2[0], 0]
  Q1 = P1[n/2:]
  R1 = P1[:n/2]
  Q2 = P2[n/2:]
  R2 = P1[:n/2]
  A = Karatsuba(Q1, Q2, n/2)
  B = Karatsuba(R1, R2, n/2)
  C = Karatsuba(Q1 - R1, R2 - Q2, n/2)
  for i in range(n):
    S[i] = B[i]
  for i in range(n, 2 * n):
    S[i] = A[i]
  for i in range(n/2, 3 * n / 2):
    S[i] += A[i] + B[i] + C[i]
  return S
```

En complexite:

$T(n) = 3T(\frac{n}{2} + \Theta(n)$
$T(n) = \Theta(1)$