

Proiectarea algoritmilor

Lucrare de laborator nr. 9

Paradigma *Divide_et_Impera*

Înmulțirea numerelor întregi mari

Cuprins

1	Algoritm <i>Divide_et_Impera</i> clasic	1
2	Algoritm <i>Divide_et_Impera</i> îmbunătățit	1
3	Sarcini de lucru și barem de notare	2

1 Algoritm *Divide_et_Impera* clasic

Să presupunem că dorim să înmulțim două numere întregi x și y , formate din n cifre, în baza b .

Putem presupune ca x și y sunt pozitive. Algoritmul clasic necesită $O(n^2)$ operații.

De exemplu, dacă baza $b = 10$ și $x = 61, 438, 521$ și $y = 94, 736, 407$, atunci

$xy = 5, 820, 464, 730, 934, 047$.

Să spargem acum x și y în două jumătăți. Rezultă $x_s = 6, 143$, $x_d = 8, 521$, $y_s = 9, 473$, și $y_d = 6, 407$.

Vom avea $x = x_s 10^4 + x_d$ și $y = y_s 10^4 + y_d$. Urmează că $xy = x_s y_s 10^8 + (x_s y_d + x_d y_s) 10^4 + x_d y_d$.

Astfel, pentru înmulțirea numerelor întregi x și y , sunt necesare 4 înmulțiri de numere formate din $n/2$ cifre: $x_s y_s$, $x_s y_d$, $x_d y_s$ și $x_d y_d$.

Înmulțirea cu 10^8 și 10^4 înseamnă adăugarea de zerouri, ceea ce implică $O(n)$ operații suplimentare.

2 Algoritm *Divide_et_Impera* îmbunătățit

Dacă înmulțim recursiv obținem recurența $T(n) = 4T(n/2) + O(n)$

Din teorema complexității *Divide_et_Impera* rezultă $T(n) = O(n^2)$.

Pentru a obține un algoritm subpătratic, trebuie să reducem numărul apelurilor recursive.

Observația cheie este $x_s y_d + x_d y_s = (x_s - x_d)(y_d - y_s) + x_s y_s + x_d y_d$

În locul a două înmulțiri pentru a obține coeficientul lui 10^4 , putem face o înmulțire și apoi să folosim rezultatul a două înmulțiri deja efectuate.

Astfel, pentru înmulțirea numerelor întregi x și y , sunt necesare 3 înmulțiri de numere formate din $n/2$ cifre: $x_s y_s$, $x_d y_d$ și $(x_s - x_d)(y_d - y_s)$.

În acest fel, numărul apelurilor recursive este redus la 3.

Rezultă $T(n) = 3T(n/2) + O(n)$, adică $T(n) = O(n^{\log_2 3}) = O(n^{1.59})$.

3 Sarcini de lucru și barem de notare

Sarcini de lucru:

1. Scrieți un program C/C++ care implementează o metodă *Divide_et_Impera* pentru înmulțirea a două numere întregi formate din 128 cifre binare.

Barem de notare:

1. Aplicarea corectă a metodei *Divide_et_Impera*: 6p
2. Funcția de înmulțire conține 3 apeluri recursive: 3p
3. Baza: 1p

Bibliografie

- [1] M. A. Weiss, *Data Structures and Algorithm Analysis in C*, The Benjamin/Cummings Publishing Company, Inc., 1992.