Structuri de date si algoritmi (tema 2-51)

Exercitiul 1.

Descrieți o reprezentare pentru numere întregi care să nu aibă nicio restricție legată de valoarea maximă ce poate fi reprezentată (în afară de restricțiile legate de resursele de memorie ale sistemului de calcul). Descrieți pe scurt cum poate fi folosită această reprezentare pentru calcule de adunare, înmulțire și ridicare la putere.

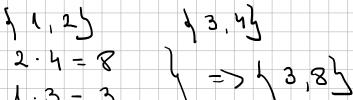
O astfel de reprezentare ar putea fi realizata folosind o structura de date dinamica(un vector de exemplu) pentru a stoca cifrele numarului.

De exemplu, numărul 12345 poate fi stocat ca {1, 2, 3, 4, 5}. Un bit sau un element poate fi folosit pentru a stoca semnul numărului (pozitiv sau negativ).

Operatiile:

- Adunare : Se aduna fiecare bloc corespondent pentru doua numere si se propaga transportul daca este necesar, iar operatia se continua pana la ultimul bloc.

-Inmultire: Se inmulteste fiecare bloc din primul numar cu fiecare bloc din al doilea numar(ca metoda clasica de inmultire pe hartie). Din nou se propaga transportul si se continua pana la ultimul bloc in acelasi fel.



-Ridicare la putere: Putem folos algoritmul de exponentiere rapida, adica se impart calculele in inmultiri repetate.

In cazul unui exponent par, putem inmulti doua puteri identice (pentru 2^8 ---> 2^4 * 2^4 <=> (2^2 * 2^2) * (2^2 * 2^2) si tot asa. In cazul unui exponent impar, se mai adauga inca o inmultire suplimentara.

Exercitiul 2

Ex2. Definiți un TDA listă de întregi. Specificați ce operații doriți să fie definite pentru acest TDA. Apoi propuneti un fișier de tip header pentru o implementare a unei biblioteci in C pentru TDA abstract definit anterior.

typedef struct Lista {
 int *elemente;
 int dimensiune_curenta;
 int capacitate;
}Lista_t;