Feladat

Valósítsa meg a nagyon nagy természetes számok típusát! Ábrázoljuk a számokat a számjegyeik láncolt listáival és implementálja az összeadást és a szorzást operátor túldefiniálással! Ne feledkezzen meg a beolvasó (operátor>>) és kiíró (operátor<<) metódusokról sem! Az összeadás műveletigénye Ơ(m+n), a szorzásé Ơ(m*n), ahol m és n a két szám számjegyeinek száma.

Bignum típus

A feladat megoldásához definiálunk egy bármilyen nagy természetes számokat tartalmazó típust.

Típusérték-halmaz

Számjegyek sorozata: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}*

Típus-műveletek

A típushoz az alábbi műveleteket vezetjük be:

<u>Módosító-műveletek:</u>

- linkLeftOf(ptr) : adott elemtől(ptr) balra beszúrás
- linRightOf(ptr) : adott elemtől(ptr) jobbra beszúrás

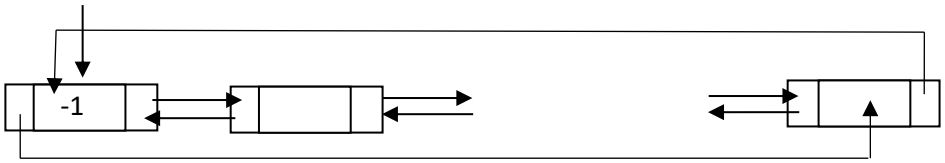
<u>Operátorok:</u>

- *+* : két Szám típusú objektum összeadása $\in O(m+n)$
- *** : két Szám típusú objektum összeszorzása $\in O(n^{log_2 3})$

<u>Bejáró műveletek</u>

- first(): rááll a sorozat első elemére
- next(): rááll a sorozat következő elemére, ha van
- current(): visszaadja az aktuális elemét a sorozatnak
- end(): jelzi, ha már nincs következő elem

Bárdosi Bence VY9NJN - Dokumentáció 2. beadandóhoz

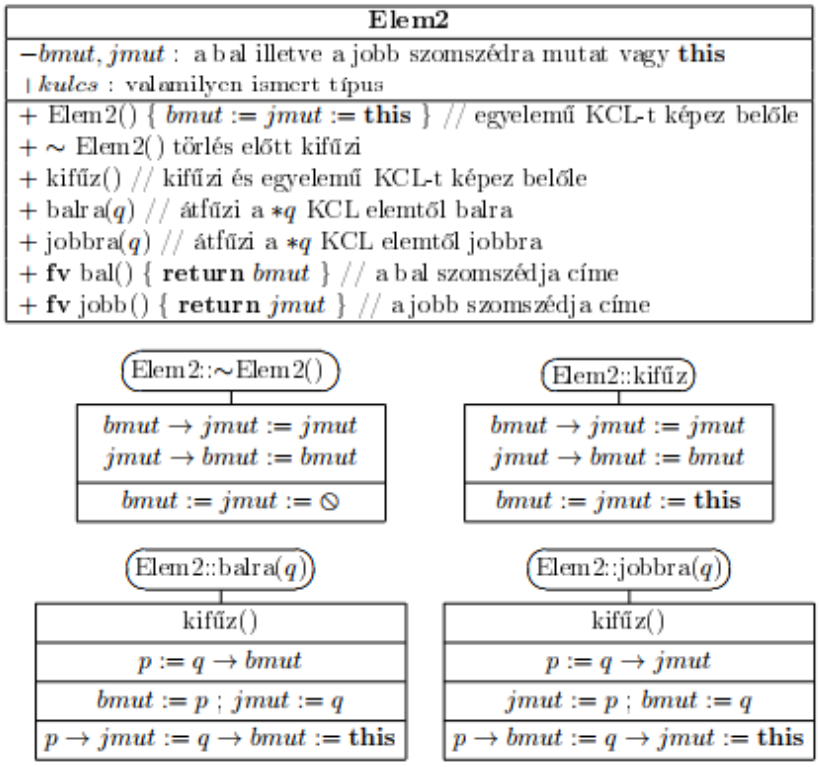


Egy sorozatot fejelemes ciklikus kétirányú láncolt listával fogunk ábrázolni. Egy listaelem mezői: **pLeft** (pointer), **value** (számjegy), **pRight** (pointer). A láncolt listát az alábbi pointerek azonosítják:

- **first**: lista fejelemére mutat

Fontos invariáns tulajdonság, hogy az elemek mutatói sosem lehetnek nullpointerek.

Implementáció



Algoritmus ~ Programkód

Elem2	Node
bmut, jmut	pLeft, pRight
kulcs	value
kifűz	unlink
balra/jobbra	ink(Left/Right)Of
bal, jobb	left, right

Megoldás C++-ban

Bignum-típus

A Bignum típust egy osztállyal valósítjuk meg. A Bignum osztály deklarációját a Bignum.h fejállományban helyezzük el, a metódusok implementációit a Bignum.cpp és multiply.cpp forrásállományban. Az osztály publikus része a típusspecifikációban felsorolt műveleteken kívül a konstruktort, a destruktort, a másoló konstruktort és az értékadás operátort tartalmazza. A sorozat-típus bejáróit egy külön osztály, a Bignum osztályba ágyazott Enum osztály írja leírja le.

A hibakezelésre két kivételt definiálunk. A NotNaturalNum kivétel a beolvasásnál aktiválódik, ha a beolvasásnál egy olyan karaktert akarunk beolvasni ami nem szám (nincs benne a {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} halmazban). A StratsWithZero pedig ha a szám nem 0, de ’0’ karakterrel kezdődik.

Tesztelési terv

1) Bignum konstruktor
    a. Int típusú paraméterrel
    b. String típusú paraméterrel
2) Másoló konstruktor és értékadás operátor
3) Összeadás
    a. 0-elemmel (0) való összeadás
    b. Kommutativitás
    c. Asszociativitás
    d. két sizeof(int)-nél nagyságrendekkel nagyobb szám összege
    e. két random szám összege 10szer
4) Szorzás
    a. Szorzás segédfüggvényei
    b. 0-val szorzás
    c. 0-elemmel (1) szorzás
    d. kommutativitás
    e. asszociativitás
    f. két sizeof(int)-nél nagyságrendekkel nagyobb szám szorzata
    g. két random szám szorzata 10szer
5) Streamkezelő operátorok (<< >>)
    a. <<
    b. >>
6) Kivételkezelés
    a. NotNaturalNum
    b. StartsWithZero