

2. Beadandó feladat dokumentáció

Készítette:

Giachetta Roberto

EHA: GIRIAAT.ELTE

E-mail: groberto@inf.elte.hu

Feladat: Valósítsuk meg a prioritásos sor típusát, amelynek elemei prioritás/érték párok, és mindig a legnagyobb prioritású elemet kérdezhajjuk le, illetve vehetjük ki a sorból. A sor tároljon egész prioritás, és szöveg érték típusokat. A főprogramban használjuk a prioritásos sort arra, hogy a megadott szavakat hosszúság szerint csökkenő sorrendben írja vissza. A program legyen képes tetszőleges számú szót elfogadni.

Elemzés:

- A feladat megoldásához a szavakhoz a hosszukat rendeljük prioritásként, így a prioritásos sorból pont a megfelelő sorrendben vehetjük ki a szavakat. A főprogramban létrehozunk a sort, feltöltjük szavakkal, majd törlés közben kiírjuk azokat.
- A prioritásos sor megszokott műveletei a beszúrás, maximális prioritású elem lekérdezése és kivétele. A sor elemeit rendezetlenül rakjuk be, így a kivétel esetén újra kell keresnünk a maximális prioritású elemet. A sor műveletei:
 - beszúrás (*Insert*)
 - maximális elem lekérdezése (*Max*)
 - maximális elem kivétele (*DelMax*)
 - sor kiürítése (*Clear*)
 - sor hosszának lekérdezése (*Length*)
- Tetszőlegesen sok elem fogadásához a sort láncolt listával reprezentáljuk, pontosabban fejelemes, egyirányú, aciklikus listát.
- A prioritásos sor rendezetlenül tárolja az elemeket, valamint mindig tárolja a maximális elem helyét, így a beszúrás és a maximum elem lekérdezés hatékonyan ($\theta(1)$ műveletigény mellett) történik, viszont a kivételkor újra kell keresni a maximális elemet ($\theta(n)$ műveletigény mellett).

Specifikáció:

Prioritásos sor típus:

Típus specifikáció: (*PriQueue*, {*Insert*, *Max*, *DelMax*, *Clear*, *Length*}), ahol

$PriQueue \subset (\mathbb{Z} \times \mathbb{Ch}^*)^*$,

$Insert: PriQueue \times \mathbb{Z} \times \mathbb{Ch}^* \rightarrow PriQueue$,

$Max: PriQueue \rightarrow \mathbb{Ch}^*$,

$DelMax: PriQueue \rightarrow \mathbb{Ch}^* \times PriQueue$,

$Clear: PriQueue \rightarrow PriQueue$,

$Length: PriQueue \rightarrow \mathbb{N}$

Típus realizáció (informális):

- A prioritásos sor tartalmazza a lista fejelemét, a maximális elem mutatóját, valamint az elemek számát (a hatékony elemszám lekérdezés végett).
- Új elem beszúrásakor az a lista elejére kerül, és prioritásának függvényében változik a maximális mutató értéke (illetve ha korábban üres volt a lista).

- A maximális elem lekérdezése a maximális elem mutatóján keresztül történik, törlésekor kivesszük a maximális elemet a listából, és újra beállítjuk a mutató értékét maximumkereséssel.
- A hatékony hossz lekérdezés érdekében eltároljuk a prioritásos sor hosszát is.
- A típus számára definiáljuk az üres sor kivételét (*PRIQUEUE_IS_EMPTY*), amelyet a maximális elem lekérdezésekor és kivételekor dobhat a típus.

Implementáció:

- A prioritásos sort (*PriQueue*) külön modulban implementáljuk: *priqueue*.
- A listához külön elemtípust definiálunk (*Element*), amelyet beágyazott típusként valósítunk meg, és tartalmazni fogja a kulcs és érték mellett a következő elem mutatóját.
- A típus megfelelő működéséhez megvalósítjuk a másoló konstruktor, értékadás és destruktorkonstruktor műveleteket is. A maximumkeresés műveletét külön metódusba helyezzük.

Tesztelés:

- Elemek beszúrása a prioritásos sorba tetszőleges prioritással, hosszának megfelelő prioritással. Halmazos beszúrás (egymást követően 1000000 adatpár).
- Maximum kivétel és lekérdezés nem üres, valamint üres sorból.
- Értékadó operátor, másoló konstruktor ellenőrzése.