Karakterláncok – Boyer-Moore

-specifikációk-

A karakterláncok olyan karaktertömbök, amelyek segítségével egy karaktersorozatot tudunk eltárolni egyetlen karaktertömbben.

**Tulajdonságok:**

* homogén
* O(n \* m) idő bonyolultságú, ahol n a forrás szöveg hossza, az m pedig az ebben keresendő mintáé
* bármilyen karakterekből álló karakterláncot képes eltárólni
* mérete dinamikus, lehet egyaránt növelni és csökkenteni
* az adatszerkezet neve a programban: „str\_bm”
* adattagok: „char\* data” típusú változó, melyben maga az adat és „int len” melyben a karakterlánc hossza van eltárolva

**Műveletek:**

* **Létrehozás:**

1. üresen:

Előf.: -

Utóf.: létrejön egy üres karakterlánc

1. értékadással:

Előf.: megadjuk paraméterként a k.l. tartalmát const char\* vagy str\_bm típusként

Utóf.: létrejön a karakterlánc a megadott adattal

* **Felszabadítás:**

felszabadítja az adatszerkezet által lefoglalt memóriát, amennyiben az létezik

* **get\_data():**

visszatéríti a karakterlánc tartalmát egy const char\* típusú változóban

Előf.: -

Utóf: -

* **strlen:**

visszatéríti egy paraméterként megadott const char\* típusú karakterlánc hosszát

* **bad\_char:**

inicializál egy 256 hosszú vektort egy paraméterként megadott minta alapján a rossz karakter szabályt követve:

pl. minta = “alma”, akkor a bc[‘a’] = 3, bc[‘l’] = 1, bc[‘m’] = 2

* **&operator[]:**

visszatérít egy megadott indexen levő elemet

Előf.: a megadott index valódi

Utóf: visszatérití az adott indexen levő karaktert

* **&operator=:**

Előf.: paraméter legyen ugyanolyan típusú

Utóf.: egy k.l. értékét megváltoztatja egy másikéra

pl. Kar a = b + c

* **operator+:**

Előf.: paraméter legyen ugyanolyan típusú

Utóf.: visszatéríti 2 k.l. összefűzött értékét

pl. Kar a = b + c

* **&operator+=:**

Előf.: paraméter legyen ugyanolyan típusú

Utóf.: hozzáfűzi a k.l.-hoz a paraméter tartalmát

pl. Kar a += b

* **to\_lower:**

kisbetűre állítja a nagybetűket a k.l.-ban

Előf.: -

Utóf.: -

* **to\_upper:**

nagybetűre állítja a kisbetűket a k.l.-ban

Előf.: -

Utóf.: -

* **&operator>>:**

megadott streamről olvas be, fehér karakterig

Előf.: -

Utóf.: -

* **&operator<<:**

megadott streamre kiirja a karakterláncot

Előf.: -

Utóf.: -

* **substr:**

visszatéríti a karakterlánc egy részkarakterláncát

Előf.: megadunk 2 valódi indexet (intervallumot)

Utóf.: visszatéríti a megadott részét a k.l.-nak

* **insert:**

beszúr a k.l.-ba egy másikat

Előf.: a pozíció valódi

Utóf.: beszúrja a megadott helyre a k.l.-ot

* **replace:**

kicserél egy megadott részt a karakterláncban

Előf.: a megadott index valódi

Utóf.: kicseréli a megadott helyen levő részt

* **reverse:**

visszatérití a k.l. fordítottját

Előf.: -

Utóf.: -

* **lenght:**

visszatéríti a karakterlánc hosszát egy int típusú változóban

Előf.: -

Utóf: -

* **find\_from:**

megkeresi a paraméterként megadott minta összes előfordulásat, melyeket eltárol egy vektorban

ha nem talál előfordulást, akkor a vektor üres marad

Előf.: -

Utóf.: -

* **find\_from\_see\_steps:**

ugyanaz, mint az előző, csak közben nyomon követhetjük a változók értékét a terminalban

* **find\_from\_naive:**

a fentebb implementált karakterlánc kereső naiv változata

* **read\_from:**

beolvas fájlból

Előf.: a fájl paraméter const char\* típusú és a fájl létezik

Utóf.: sikeres beolvasás

* **get\_line:**

újsor karakterig olvas egy megadott streamről

Előf.: -

Utóf.: sikeres beolvasás