Alga2

Halmaz

Halmaz: Azok a halmazok, amelyek bővülhetnek, zsugorodhatnak dinamikus halmaznak nevezzük. Ezeken a halmazokon 2 fajta műveletet különböztetünk meg lekérdező, illetve módosító műveletek.

-lekérdező művelet: keres, minimum, maximum, előrző, következő

-módosító művelet: beszúr, töröl

Verem

Verem: A verem egy spciális halmaz, amely egy n elemű tömbbel megvalósítható. LIFO szerkezet vonatkozik rá. Tudunk a verembe beszúrni és törölni is, ilyenkor beszúráskor a sor végére kerül az elem, törléskor (kivétel) pedig a legutóbb beszúrt elemet töröljük.

Sor: FIFO szerkezetű. Végrehajtható rajta a sorba beszúrás és sorból törlés. Beszúráskor a sor végére kerül az elem, törléskor (kivétel) pedig azt az elemet vesszük ki amelyik már legrégebb óta a sorban van.

Láncolt lista

Láncolt lista: A láncolt lista elemei referenciákat tartalmaznak. Referenciák típusa alapján megkülönböztetünk kétfajta lácolt listát:

-egyszeresen láncolt: csak a következő elemre tartalmaz hivatkozást

-kétszeresen láncolt: az előző és a következő elemre is tartalmaz hivatkozást

Számos további változata lehet egy láncolt listának:

-ciklikus lista: az első elem előzője az utolsó és az utolsó elem következője pedig az első

-nem ciklikus: az első elem előzője NIL és az utolsó elem következője is NIL

-rendezett: valamilyen prioritás szerint rendezve van

-nem rendezett: nincs prioritási sorrend az eleme között

Bináris keresőfák

Minden művelet futásideje logn. Bináris kereső fa legfőbb tulajdonsága, hogy vegyük egy csúcsnak a részfáját jobb és baloldali részfáját, ahol a baloldali mindig kisebb lesz a jobb pedig mindig nagyobb/egyenlő mint a csúcs érték a fában (szülő). Itt az a lényeg hogy a fa magassága legrosszabb esetben is n legyen és ne az elemek számától, hanem a fa magasságától függjön.

Műveletek:

-keresés: gyökérelemtől kezdve vizsgáljuk, hogy a fa csúcsában található érték megegyezik-e a keresett értékkel, ha nem akkor annak függvényében, hogy a keresett érték kisebb-e vagy nagyobb folytatjuk a keresést a gyerekénél.

-min/max: Addig használjuk a keresést amíg meg nem találjuk azt az elemet a fában amelyiknek már nincs gyereke. Min esetén a gyökérnek a bal gyerekeit vizsgáljuk, max esetén pedig a jobb oldalt.

következő/előző: Ilyenkor van egy elemünk x amihez képest kell megtalálni a következőt/előzőt. Következő esetén a keresést végrehajtjuk a fában és megvizsgáljuk, hogy az x-nek van-e jobb gyereke, ha van akkor ebben a részfában minimumot keresünk és visszaadjuk azt mint következőt. Ha viszont nincs jobb gyerek x-nek akkor addig fejtjük vissza a szülőket, ameddig nem találjuk meg azt a számot amelyik nagyobb mint x.

Előző esetén a keresést végrehajtjuk a fában és megvizsgáljuk, hogy az x-nek van -e bal gyereke, ha van akkor ebben a részfában keresünk maximumot és visszaadjuk azt, mint előzőt. ha viszont nincs bal gyereke x-nek akkor addig fejtjük vissza a szülőket, ameddig nem találjuk meg azt a számot amelyik kisebb mint x.

-beszúrás: Meg kell találni azt a helyet ahova beszúrható az elem. Tehát az aktuális csúcsnál nagyobb az érték, akkor megnézzük, hogy a jobb gyereke szabad-e, ha igen akkor beszúrjuk, ha nem , akkor a jobb gyerekén folytatjuk ezt az eljárást. Fa magasságtól függ a futásidő O(n).

-törlés: Törlésnél, ha a törlendő érték levél akkor egyszerűen kitöröljök. Ha egy gyerekkel rendelkezik akkor egyszerűen összekötjük a szülőt a tárolt csúcs gyerekeivel. Ha két gyerekkel rendelkezik, akokr a bal gyerekénak a legnagyobb gyerekét tesszük a helyére.

-inorder bejárás: először bejárjuk bal gyerekből képzett részfát majd a jobb gyerekből képzettet is.

Hasító táblák: (futásidő: legrosszabb O(n), legjobb O(1))

A hasító táblák egy nagyon hatékony adatszerkezetek, amelyek kulcs-érték párokat tárolnak. Az adatok elérését és tárolását a hash függvények segítségével végzik. Egy k kulcsú elem a k-adik résben tárolódik. A hasítás alkalmazása esetén az elem h(k) helyre kerül vagyis a hasító függvényt használjuk arra, hogy a rést a k kulcsból meghatározzuk. A hasító függvény célja, hogy csökkentse a szükséges tömbindexek tartományát.

Ütközés feloldás láncolással: az ugyanarra a résre mutató elemeket összefogjuk egy láncolt listába.

Gráfok és fák számítógépes reprezentációja

fák ábrázolása

Csúcsokat és éleket kell reprezentálni ezek a lehetőségek vannak :

-gyerek éllista: tartalmazza a kulcsot, a szülőt és a leszármazottait listában

-első fiú, apa, testvér: tartalmazza a kulcsot, a szülőt, a gyereket és egy testvért

-bináris fa: tartalmazza a szülőt, a kulcsot, a jobb és bal gyereket

gráfok ábrázolás

-irányított és irányítatlan gráfokat ugyanúgy ábrázoljuk csak más szabályok szerint

-szomszédsági lista: Irányítatlan esetén, ha két csúcs között van él akkor azt felvesszük. Irányított esetén, ha a csúcsba vezet él akkor azt felvesszük.

-csúcsmátrix: Irányítatlan gráf esetén mindkét csúcs esetén 1-et írunk a csúcskhoz. Irányítatlan esetén pedig csak ahhoz a csúcshoz írunk 1-et, ahova érkezik az él.