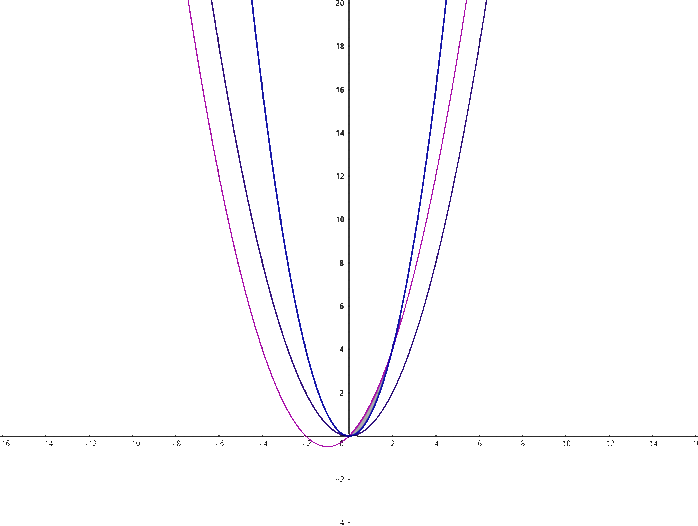
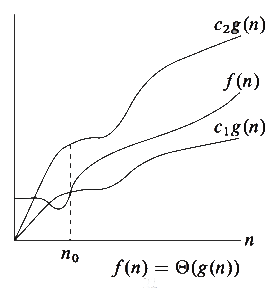
| Adatszerkezetek  és algoritmusok | Aszimptotikus korlát |
| --- | --- |
| 3. ábra Felső korlát | Aszimptotikus korlát Az aszimptotikus korlát segítségével el lehet dönteni, egy adott algoritmusnak mennyi az időigénye. Aszimptotikus felső korlát Egy adott g(n) függvény esetén O(g(n)) jelöli a függvényeknek azt a halmazát, amelyre O(g(n)) = {f(n): Létezik c és n0 pozitív állandó úgy, hogy 0 <= f(n) <= c\*g(n) teljesül minden n>=n0 esetén}  Tehát létezik egy olyan n0 pont, amitől a g(n) függvény csak nagyobb értékeket fog felvenni. Minta feladat 3\*n+8=O(n) esetén meghatározunk egy c számot-> a legnagyobb növekedésű tag szorzójához hozzáadunk egyet, ebben az esetben c=4, ezután megnézzük hogy melyik érték estén lesz nagyobb a korlát, mint az értékek, erre készítünk egy táblázatot.  Most elkészítjük a 2 függvényünket, az első a g(n)=3\*n+8, ez a képlet adott volt. Ezután elkészítjük a másik képletünket f(n)=4\*n, ahol a 4 a c-ként megadott érték.   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | g(n) | 11 | 14 | 17 | 20 | 23 | 26 | 29 | 32 | 35 | 38 | | f(n) | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 | |  |  | | |  | | |  | | | | |  | f(1)=4\*1=4  f(2)=4\*2=8  f(4)=4\*3=12  f(4)=4\*4=16  f(5)=4\*5=20  f(6)=4\*6=24  f(7)=4\*7=28  f(8)=4\*8=32  f(9)=4\*9=36  f(10)=4\*10=40 | | | g(1)=3\*1+8=11  g(2)=3\*2+8=14  g(3)=3\*3+8=17  g(4)=3\*4+8=20  g(5)=3\*5+8=23  g(6)=3\*6+8=26  g(7)=3\*7+8=29  g(8)=3\*8+8=32  g(9)=3\*9+8=35  g(10)=3\*10+8=38 | | |  | | | | |  |  | | |  | | |  | | | |   Ebben az esetben a látjuk, hogy n0=8, mivel g(n) és f(n) értéke is 32 tehát megegyezik. Ebből az állításból arra következteünk, hogy 4\*n nagyobb, egyenlő 3\*n+8, ha n>=8.            Aszimptotikus alsó korlát Egy adott g(n) függvény esetén Ω(g(n)) jelöli a függvényeknek azt a halmazát, amelyre Ω(g(n)) = {f(n): esetén létezik c és n0 pozitív állandó úgy, hogy 0<=c\*g(n)<=f(n) teljesül minden n>=n0 esetén.}  Tehát létezik egy olyan n0 pont, amitől a g(n) függvény csak kisebb értékeket fog felvenni. Minta feladat 2\*n-6=O(n) esetén meghatározunk egy c számot-> a legnagyobb növekedésű tag szorzójából kivonunk egyet, ebben az esetben c=1, ezután megnézzük hogy melyik érték estén lesz kisebb a korlát, mint az értékek, erre készítünk egy táblázatot.  Most elkészítjük a 2 függvényünket, az első a g(n)=2\*n-6, ez a képlet adott volt. Ezután elkészítjük a másik képletünket f(n)=n, ahol a 1 a c-ként megadott érték.   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | g(n) | -4 | -2 | 0 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | | f(n) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |  |  | | |  | | |  | | | | |  | f(1)=1\*1=1  f(2)=1\*2=2  f(3)=1\*3=3  f(4)=1\*4=4  f(5)=1\*5=5  f(6)=1\*6=6  f(7)=1\*7=7  f(8)=1\*8=8  f(9)=1\*9=9  f(10)=1\*10=10 | | | g(1)=2\*1-6=-4  g(2)=2\*2-6=-2  g(2)=2\*3-6=0  g(4)=2\*4-6=2  g(5)=2\*5-6=4  g(6)=2\*6-6=6  g(7)=2\*7-6=8  g(8)=2\*8-6=10  g(9)=2\*9-6=12  g(10)=2\*10-6=14 | | |  | | | | |  |  | | |  | | |  | | | |   Ebben az esetben a látjuk, hogy n0=6, mivel g(n) és f(n) értéke is 6 tehát megegyezik. Ebből az állításból arra következteünk, hogy n kisebb, egyenlő 2\*n-6, ha n>=6.  2. ábra Alsó korlát             Aszimptotikus éles korlát Egy adott g(n) függvény esetén Θ(g(n)) jelöli a függvényeknek azt a halmazát, amelyre Θ(g(n)) = {f(n): létezol c1, c2 és n0 pozitív állandó úgy, hogy 0 <= c1\*g(n) <= c2\*g(n) teljesül minden n>=n0 esetén} Minta feladat n2/2+n =Θ(n2) esetén meg kell határozni c1 és c2 értékét. Legyen c1=0,5 és c2=1. Ekkor a következő képeleteket fogjuk megkapni:  g(n) = n2/2+n  f(n)=0,5\*n2  b(n)=1\*n2   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  | |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | b(n) | 1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 81 | 100 | | g(n) | 1,5 | 4 | 7,5 | 12 | 17,5 | 24 | 31,5 | 40 | 49,5 | 60 | | f(n) | 0,5 | 2 | 4,5 | 8 | 12,5 | 18 | 24,5 | 32 | 40,5 | 50 | |  |  | | |  | | |  | | | | |  | b(1)=1^2=1  b(2)=2^2=4  b(3)=3^2=8  b(4)=4^2=16  b(5)=5^2=25  b(6)=6^2=36  b(7)=7^2=49  b(8)=8^2=64  b(9)=9^2=81  b(10)=10^2=100 | | | g(1)=1^2/2+1=1,5  g(2)=2^2/2+2=4  g(3)=3^2/2+3=7,5  g(4)=4^2/2+4=12  g(5)=5^2/2+5=17,5  g(6)=6^2/2+6=24  g(7)=7^2/2+7=31,5  g(8)=8^2/2+8=40  g(9)=9^2/2+9=49,5  g(10)=10^2/2+10=60 | | | f(1)\*0,5=1^2\*0,5=0,5  f(2)\*0,5=2^2\*0,5=2  f(3)\*0,5=3^2\*0,5=4,5  f(4)\*0,5=4^2\*0,5=8  f(5)\*0,5=5^2\*0,5=12,5  f(6)\*0,5=6^2\*0,5=18  f(7)\*0,5=7^2\*0,5=24,5  f(8)\*0,5=8^2\*0,5=32  f(9)\*0,5=9^2\*0,5=40,5  f(10)\*0,5=10^2\*0,5=50 | | | | |  |  | | |  | | |  | | | | |  |  | | |  | | |  | | | |   Ekkor azt tapasztajuk, hogy b(n)<=g(n)<=f(n), ha n0=2. Tehát n>=2 |
|  |  |



4. ábra Minta feladat



1. ábra Éles korlát

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | Multihalmazok halmazműveletei |
| 5. ábra Unió  6. ábra Metszet  7. ábra B \ A  8. ábra A \ B | Multihalmaz A matematikában a multihalmaz a halmaz olyan módosítása, melyben azonos elemek többször is előfordulhatnak. HalmazműveletekUnió Az A és a B uniója (vagy egyesítése) az A ∪ B = {x: x∊A és x∊B} halmaz.  Multihalmaz esetén MAX(A,B) Metszet Az A és a B metszete (vagy közös része) az A ∩ B = {x: x∊A és x∊B} halmaz  Multihalmaz esetén MIN(A,B) Különbség Az A és a B halmaz különbsége az A-B = { x: x∊A és x∉B} halmaz}  Multihalmaz estén A[i] – B[i], ahol a – nem minuszt jelöl Minta feladat A = {1, 1, 1, 3, 3, 4, 6, 7, 7}  B= {1, 3, 3, 3, 4, 8, 9}  Felírjuk a tábláztatot, a legkisebb elemtől a legnagyobbig. Jelen esetben a legkisebb az 1 és a legnagyobb a 9.   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | A | 3 | 0 | 2 | 1 | 0 | 1 | 2 | 0 | 0 | | B | 1 | 0 | 3 | 1 | 0 | 0 | 0 | 1 | 1 | | A∪B | 3 | 0 | 3 | 1 | 0 | 1 | 2 | 1 | 1 | | A∩B | 1 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | | A \ B | 2 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | | B \ A | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | |
|  |  |

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | Mátrix reprezentációja |
| A feladatban a következő mátrix van reprezentálva:   |  |  |  |  | | --- | --- | --- | --- | | 1 | 0 | 0 | 8 | | 0 | 4 | 0 | 0 | | 0 | 0 | 0 | 0 | | 4 | 0 | 7 | 0 | | Mátrix A kétdimenziós tömböt mátrixnak hívjuk, és minden elemet 2 számjegyből álló sorozattal azonosítunk. ReprezentációiSorfolytonos reprezentáció Egy mátrix sorfolytonos reprezentációja esetén megmarad az elemekhez történő közvetlen hozzáférés. A mátrix méretét tárolni kell! Oszlopfolytonos reprezentáció Egy mátrix oszlopfolytonos reprezentációja esetén megmarad az elemekhez történő közvetlen hozzáférés. A mátrix méretét tárolni kell! Háromsoros reprezentáció A 3 soros reprezentáció alkalmazásával elveszik az elemekhez történő közvetlen hozzáférés. Csak azokat az elemeket kell megtartani ami nem 0. A mátrix méretét tárolni kell! Minta feladat Sorfolytonos reprezentáció   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1. sor | | | | 2. sor | | | | 3. sor | | | | 4. sor | | | | | 1 | 0 | 0 | 8 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 7 | 0 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |   Oszlopfolytonos reprezentáció   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1. oszlop | | | | 2. oszlop | | | | 3. oszlop | | | | 4. oszlop | | | | | 1 | 0 | 0 | 4 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 7 | 8 | 0 | 0 | 0 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |   Háromsoros reprezentáció   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Sor | 1 | 1 | 2 | 4 | 4 | | Oszlop | 1 | 4 | 2 | 1 | 3 | | Érték | 1 | 8 | 4 | 4 | 7 |   Ebben az esetben az 1. sor 1. oszlopának értéke 1, az 1. sor 4. oszlopának értéke 8. STB. |
|  |  |

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | Verem reprezentációja |
|  | Verem A verem olyan speciális absztrakt adattípus, melyen mindössze két módosító műveletet használhatunk! Mivel ezekkel a műveletekkel mindig a verembe legutoljára bekerült elem fog először kikerülni, ezért szokás LIFO (last in, first out) adattípusnak nevezni. MűveleteiPUSH Ami elemek a verem tetejéhez történő hozzáadására szolgál POP Ami a verem tetején elhelyezkedő elem hozzáférésére és egyben eltávolítására szolgál Reprezentáció A reprezentációhoz tartozik még egy érték, ami a S.Top, ez az érték megmondja, hogy a verem az adott időpontban hány elemet tartalmaz.   * Ha az S.Top = 0, akkor a verem üres.   A veremhez történő hozzáféréskor két probléma léphet fel:   * Ha a POP műveletet szeretnénk alkalmazni üres verem esetén, - ekkor alulcsordulásról beszélünk, * Ha az S.Top érték nagyobbra nő az n értéknél, - ekkor pedig túlcsordulásról beszéhetünk.  Minta feladat Legyen S egy üres, 5 elemű verem. Reprezentálja az alábbi műveletek S – en tett változásait!   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | S.Top | Return | | PUSH(s,8) | 8 | - | - | - | - | 1 | - | | PUSH(s,6) | 8 | 6 | - | - | - | 2 | - | | POP(s) | 8 | - | - | - | - | 1 | 6 | | PUSH(s,5) | 8 | 5 | - | - | - | 2 | - | | PUSH(s,1) | 8 | 5 | 1 | - | - | 3 | - | | PUSH(s,9) | 8 | 5 | 1 | 9 | - | 4 | - | | POP(s) | 8 | 5 | 1 | - | - | 3 | 9 | | PUSH(s,2) | 8 | 5 | 1 | 2 | - | 4 | - | | PUSH(s,4) | 8 | 5 | 1 | 2 | 4 | 5 | - | |
|  | A tömb betelt, abban az esetben, ha hozzáakarnánk adni még egy elemet a tömbhöz, akkor túlcsordulásról beszélnénk. (OVERFLOW)  Ha viszon ki akarnánk venni 6 elemet, akkor mivel nem tudnánk a 6.at kivenni, így alulcsordulásról beszélnénk. (UNDERFLOW) |

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | Sor reprezentációja |
|  | Sor A sor olyan speciális absztrakt adattípus, melyen mindössze két módosító műveletet használhatunk! Mivel ezen műveletek alkalmazásával mindig a sorba hamarabb bekerült elem fog először kikerülni, ezért szokás FIFO (first in, first out) adattípusnak nevezni. A sor esetén két természetes számot szükséges nyilvántartani. Az egyik a fej (HEAD), a másik a farok (TAIL). Amikor az elem bekerül a sorba, akkor mindig a farok által megadott pozícióba kerül, míg törléskor mindig a fej által megadott pozícióban lévő elem fog törlődni. MűveleteiENQUEUE Ami elemeknek a sor végére történő beszúrására szolgál DEQUEUE Ami a sor elején lévő elemhez történő hozzáférésére és egyben az adott elem eltávolítására szolgál Reprezentáció A reprezentációra a ciklikus sor a leghatékonyabb, és legelterjedtebb módszer.   * Ha a HEAD és a TAIL megegyezik, akkor a sor üres.  Minta feladat Legyen Q egy üres, 6 elemű sor. Reprezentálja az alábbi műveletek Q – n tett változásait!   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | 6 | Head | Tail | Return | | ENQUEUE(Q,4) | 4 | - | - | - | - | - | 1 | 2 | - | | DEQUEUE(Q) | - | - | - | - | - | - | 2 | 2 | 4 | | ENQUEUE(Q,7) | - | 7 | - | - | - | - | 2 | 3 | - | | ENQUEUE(Q,2) | - | 7 | 2 | - | - | - | 2 | 4 | - | | ENQUEUE(Q,9) | - | 7 | 2 | 9 | - | - | 2 | 5 | - | | ENQUEUE(Q,5) | - | 7 | 2 | 9 | 5 | - | 2 | 6 | - | | ENQUEUE(Q,3) | - | 7 | 2 | 9 | 5 | 3 | 2 | 1 | - | | DEQUEUE(Q) | - | - | 2 | 9 | 5 | 3 | 3 | 1 | 7 | | ENQUEUE(Q,1) | 1 | - | 2 | 9 | 5 | 3 | 3 | 2 | - |   Head – Az a mező ahol, elkezdődnek az elemek  Tail – Az a mező, ahol az utolsó elem van +1, ha betelt a sor, akkor az első elemmel folytatódik |
|  | Ha a DEQUEUE utasítást használjuk üres sor esetén, akkor alulcsordulásról beszélünk. (OVERFLOW)  Ha a Q.Head = Q.Tail+1, azaz tele van a sor, és az ENQUEUE műveletet használjuk, akkor túlcsordulásról beszélünk. (UNDERFLOW) |

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | HASH -Tábla |
| |  |  |  |  | | --- | --- | --- | --- | | 0 |  | → | NIL | | 1 |  | → | NIL | | 2 |  | → | NIL | | 3 |  | → | NIL | | 4 |  | → | NIL | | 5 |  | → | NIL | | 6 |  | → | NIL | | 7 |  | → | NIL | | 8 |  | → | NIL |   Alapértelmezett hash-tábla: | Hash-tábla A h hash függvény az U kulcstér elemeihez rendeli egy adott T[0…m-1] hash tábla pozícióit. h: U→{0,1,…,m-1}, ahol m tipikusan jóval kisebb, mint |U|. Ütközés Ha több különböző kulcshoz is tarozhat ugyanaz a pozíció, azt ütközésnek nevezzük. Az ütközés esetére több hatékony megoldás van, hogy tüntessük el. Ütközés feloldása osztásos módszerrel Az osztásos módszer alkalmazásakor a h hash függvény a k kulcshoz az m különböző pozíció egyikét rendeli hozzá, maradékos osztás segítségével.  A hash függvény alakja h(k) = k mod m  Például, ha a hash tábla mérete m= 12, akkor a lehetséges hash értékek a 0 és 11 közötti egész számok. Ütközés feloldás láncolással Láncolás esetén az azonos hash értékekkel rendelkező elemeket egy-egy láncolt listában tároljuk. A hash-tábla j-edik pozíciójában egy mutatót találunk, mely azon láncolt lista első elemére mutat, amelyik mindazon elemeket tartalmazza, mely elemek hash értéke j. Ha nincs ilyen, akkor az értéke NILL Minta feladat Készítsen egy kulcstranszformációs (hash) táblázatot hash függvénnyel. Szúrja bele rendre az alábbi kulcs-érték párokat és ábrázolja a táblázatot! Használjon nyílt-címzés-láncolással ütközésfeloldást! Legyen mod m=9!  Beszúrandó elemek:   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | * 19, blue | | | | | 19 = 9\*2+1 | | | | | * 27, red | | | | | 27 = 9\*3 +0 | | | | | * 51, green | | | | | 51 = 9\*5 + 6 | | | | | * 13, blue | | | | | 13= 9\*1+4 | | | | | * 37, brown | | | | | 37 = 9\*4+1 | | | | | * 31, pink | | | | | 31= 9\*3+4 | | | | | * 25, orange | | | | | 25= 9\*2+7 | | | | | 0 |  | → | 27 | red | | → | NIL |  | |  |  |  | | 1 |  | → | 19 | blue | | → | 1 | brown | | → | NIL |  | | 2 |  | → | NIL |  | |  |  |  | |  |  |  | | 3 |  | → | NIL |  | |  |  |  | |  |  |  | | 4 |  | → | 13 | blue | | → | 31 | pink | | → | NIL |  | | 5 |  | → | NIL |  | |  |  |  | |  |  |  | | 6 |  | → | 51 | green | | → | NIL |  | |  |  |  | | 7 |  | → | 25 | orange | | → | NIL |  | |  |  |  | | 8 |  | → | NIL |  | |  |  |  | |  |  |  | |
|  | Ütközés feloldása nyílt címzéssel |

|  |  |
| --- | --- |
| Adatszerkezetek és algoritmusok | Fák |
|  |  |
|  |  |