

14. Mediánok és rendezett minták

Kiválasztási probléma

Bemenet: Azonos típusú (különböző) elemek $H = \{a_1, \dots, a_n\}$ halmaza, amelyeken értelmezett egy $\leq$ lineáris rendezési reláció és egy i ($1 \leq i \leq n$) index.

Kimenet: Olyan $x \in H$, hogy $|\{y : y \in H \wedge y \leq x\}| = i$.

A kimenetben szereplő x -et a H rendezett minta i -edik (legkisebb) elemének nevezzük. A középső elemet a rendezett minta **mediánjának** nevezzük. Pontosabban, az $i = \lfloor (n+1)/2 \rfloor$ -edik elem az alsó, az $i = \lceil (n+1)/2 \rceil$ -edik elem a felső medián.

Minimum és maximum egyidejű meghatározása.

```
Procedure MiniMax(Const A:Tomb; N:Longint Var i,j:Longint );
{Kimenet: A[i]=Min(A[1..N]), A[j]=Max(A[1..N])}
Var M,k:Longint; Begin
  M:=(N div 2)*2;
  If A[1] < A[2] Then Begin
    i:=1; j:=2
  Else Begin
    i:=2; j:=1
  End;
  For k:=1 To M-1 Do
    If A[k]<A[k+1] Then Begin      {párok összehasonlítása}
      If A[k]<A[i] Then j:=k;
      If A[k+1]>A[j] Then i:=k+1;
    End Else Begin
      If A[k+1]<A[i] Then i:=k+1;
      If A[k]>A[j] Then j:=k;
    End;
  If N<>M Then Begin
    If A[N]<A[i] Then i:=N;
    Else If A[N]>A[j] Then j:=N
  End;
End{MiniMax};
```

A MINIMAX algoritmus legfeljebb $3\lfloor n/2 \rfloor$ összehasonlítást végez.

Kiválasztás ismételt felosztással (iterált particionálás)

Elvi algoritmus:

Valaszt(H:Halmaz, i:Word):Elemtip;

$\{1 \leq k \leq |H|\}$

Var H_b, H_j :Halmaz; x:Elemtip;

Begin{Valaszt}

If $|H| = 1$ **Then Begin**

Valaszt:=x (* $H = \{x\}$ *);

End Else Begin;

Feloszt(H, H_b, H_j, x);

$\{H = H_b \cup H_j \cup \{x\}, \max(H_b) \leq x < \min(H_j)\}$

$q := |H_b|$;

```

If  $i \leq q$  Then
    Valaszt:=Valaszt( $H_b, i$ );
Else If  $i > q$  Then
    Valaszt:=Valaszt( $H_j, i - q - 1$ );
Else
    Valaszt:=x
End{ $|H| > 1$ };
End{Valaszt};

```

```

Function Kivalaszt (Var T : Tomb; i:Longint):Longint;
Var
    bal, jobb, f:Longint;
Function Feloszt ( bal, jobb : Longint):Longint;
Var x, E : Elemtip;
    i, j : Longint;
Begin {Feloszt}
    x:= T[jobb];
    i:=bal-1;
    For j:=bal To jobb-1 Do
        If T[j]<=x Then Begin
            i:=i+1
            E:=T[i]; T[i]:=T[j]; T[j]:=E;
        End;
        i:=i+1;
    E:=T[i]; T[i]:=T[jobb]; T[jobb]:=E;
    Feloszt:=i;
End{Feloszt};

Begin{Kivalaszt}
    bal:=1; jobb:=N;
    While bal < jobb Do Begin
        f = Feloszt (bal, jobb);
        If i=f Then Begin
            bal:=f; Break;
        End Else If i<f-bal Then
            jobb:=f-1
        Else Begin
            i:= i-(f-bal+1);
            bal:=f+1
        End
    End{while};
    Kivalaszt:=bal
End{Kivalaszt};

```

A futási idő elemzése.

$T_{lr}(n) = O(n^2)$, lásd a gyorsrendezés legrossabb esetét.

$$T_{lj}(n) = O(n).$$

Átlagos eset

Feltesszük, hogy minden bemenet egyformán valószínű.

Felosztás az f felosztó elemmel, akkor az f felosztó index lehet:

$$f = 1, 2, \dots, n-1, n$$

Feltéve, hogy $T_a(n)$ monoton növekvő függvény,

$$T_a(n) \leq \frac{1}{n} \left(\sum_{f=1}^n T_a(\max(f-1, n-f)) + O(n) \right)$$

Mivel a felosztás ideje $O(n)$. Ha n páros, akkor a $\max$ kifejezés mindkét argumentuma 1-től $n-1$ -ig kétszer szerepel az összegzésben, ha n páratlan, akkor pedig $\lfloor n/2 \rfloor$ egyszer, a többi kétszer. Tehát

$$\max(f-1, n-f) = \begin{cases} f-1 & \text{ha } f > \lceil n/2 \rceil, \\ n-f & \text{ha } f \leq \lceil n/2 \rceil. \end{cases}$$

$$T_a(n) \leq \frac{2}{n} \sum_{f=\lceil n/2 \rceil}^{n-1} T_a(f) + O(n)$$

Tegyük fel, hogy van olyan c konstans, hogy $T_a(n) \leq cn$ és hogy c teljesíti a rekurzív képlet kezdeti feltételét, azaz $T(n) = O(1)$ valamely konstansnál kisebb n -ekre. Tegyük fel továbbá, hogy a felosztás futási idejét kifejező $O(n)$ tagra teljesül, hogy $\leq an$.

$$\begin{aligned} T_a(n) &\leq \frac{2}{n} \sum_{f=\lceil n/2 \rceil}^{n-1} cf + an \\ &= \frac{2c}{n} \left(\sum_{f=1}^{n-1} f - \sum_{f=1}^{\lceil n/2 \rceil - 1} f \right) + an \\ &= \frac{2c}{n} \left(\frac{(n-1)n}{2} - \frac{(\lceil n/2 \rceil - 1) \lceil n/2 \rceil}{2} \right) + an \\ &\leq \frac{2c}{n} \left(\frac{(n-1)n}{2} - \frac{(n/2 - 2)(n/2 - 1)}{2} \right) + an \\ &= \frac{2c}{n} \left(\frac{n^2 - n}{2} - \frac{n^2/4 - 3n/2 + 2}{2} \right) + an \\ &= \frac{c}{n} \left(\frac{3n^2}{4} + \frac{n}{2} - 2 \right) + an \\ &= c \left(\frac{3n}{4} + \frac{1}{2} - \frac{2}{n} \right) + an \\ &\leq \frac{3cn}{4} + \frac{c}{2} + an \\ &= cn - \left(\frac{cn}{4} - \frac{c}{2} - an \right). \end{aligned}$$

Más csak azt kell megmutatnunk, hogy elég nagy n -ekre a fenti kifejezés legfeljebb cn , ami ekvivalens azzal, hogy $cn/4 - c/2 - an \geq 0$. Mindkét oldalhoz $c/2$ -t adva és kiemelve n -et, azt kapjuk, hogy $n(c/4 - a) \geq c/2$. Ha a c konstans úgy választjuk, hogy $c/4 - a > 0$, vagyis $c > 4a$, oszthatunk $c/4 - a$ -val, és kapjuk

$$n \geq \frac{c/2}{c/4 - a} = \frac{2c}{c - 4a}.$$

Tehát feltéve, hogy $T_a(n) = O(1)$ ha $n < 2c/(c - 4a)$, $T_a(n) = O(n)$.

Kiválasztás legrosszabb esetben lineáris időben

Alapötlet: a felosztó elem "jó" választása.

```

LinValaszt(H:Halmaz, i:Longint):Elemtip;
{1 ≤ i ≤ |H|}
Var Hb, Hj:Halmaz;
Begin{LinKiValaszt}
  If |H| = 1 Then Begin
    LinKiValaszt:=x (* H = {x} *);
  End Else Begin;
  L := N div 5;
  H = H1 + ... + HL + HL+1; {ötös csoportok}
  {|Hi| = 5, i = 1..L, |HL+1| < 5};
  Med5 := {Valaszt(H1, 3), ..., Valaszt(HL, 3)};
  Fe := LinValaszt(Med5, L/2);
  Feloszt(H, Fe, Hb, x, Hj); {felosztás az Fe felosztó elemmel}
  {max(Hb) ≤ x < min(Hj)}
  q := |Hb|;
  If i = q + 1 Then
    LinKiValaszt:=x;
  Else If i ≤ q Then
    LinKiValaszt:=LinKiValaszt(Hb, i);
  Else Begin{q + 1 < i}
    LinKiValaszt:=LinKiValaszt(Hj, i - q - 1);
  End;
End{LinKiValaszt};

```

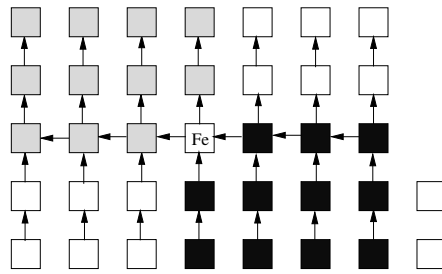
A futási idő elemzése

Képzeljük el, hogy rendeztük az ötös csoportokat, majd ezeket a középső elemük szerint. Ezt szemlélteti az 1. ábra. Az ábrán szürkére festett elemek mindegyike kisebb vagy egyenlő, mint az Fe felosztó elem, a feketék pedig nagyobbak, mint Fe .

$$q = |H_b| \geq 3 \left(\left\lceil \frac{1}{2} \left\lceil \frac{n}{5} \right\rceil \right\rceil - 2 \right) \geq \frac{3n}{10} - 6$$

Tegyük fel, hogy a legfeljebb 140 elemű bemenetekre a futási idő $\Theta(1)$. Ekkor, mivel a felosztás ideje $O(n)$, azt kapjuk, hogy

$$T_{lr}(n) \leq \begin{cases} \Theta(1) & \text{ha } n \leq 140, \\ T_{lr}(\lceil n/5 \rceil) + T_{lr}(7n/10 + 6) + O(n) & \text{ha } n > 140. \end{cases}$$



1. ábra. Az ötös csoportok szerepe a felosztásban.

Tegyük fel továbbá, hogy a felosztás futási idejét kifejező $O(n)$ tagra teljesül, hogy $\leq an$ valamely a konstansra. Tehát

$$\begin{aligned}
 T_{lr}(n) &\leq c \lceil n/5 \rceil + c(7n/10 + 6) + an \\
 &\leq cn/5 + c + 7cn/10 + 6c + an \\
 &= 9cn/10 + 7c + an \\
 &= cn + (-cn/10 + 7c + an),
 \end{aligned}$$

amely legfeljebb cn , ha

$$-cn/10 + 7c + an \leq 0. \quad (1)$$

Az (1) egyenlőtlenség teljesül, ha $c \geq 10a(n/(n-70))$ és $n > 70$. Mivel feltettük, hogy $n \geq 140$, kapjuk, hogy $n/(n-70) \leq 2$, és így $c \geq 20a$ választása esetén teljesül a kívánt egyenlőtlenség, tehát $T_{lr}(n) = O(n)$.

Megvalósítás

```

{ Globalis programelemek a LinKiv eljárashoz :
Const
    MaxN = ???                ; (* maxi. elemszám *)
Type
    Kulcstip = ???            ; (* a rendezési mező kulcstípusa *)
    Adattip = ???             ; (* az adatmező típusa *)
    Elemtip = Record
        kulcs : Kulcstip;
        adat : Adattip
    End;
    Tomb = Array[1..N] Of Elemtip;
}

Function LinKivalaszt(Var T:Tomb;N, i:Longint) : Longint;
Const
    R0 = 2;
    R = 2*R0+1;

Procedure BeszuroRend(Elso,Elszam,d: Longint);
{A rendezendő elemek T[Elso], T[Elso+d], T[Elso+d*(Elszam-1)]}
Var
    i,Utolso : Longint; E : Elemtip; j:Longint;

```

```

Begin
  i:= Elso+d; Utolso:= Elso+(Elszam-1)*d;
  While i <= Utolso Do Begin
    E := T[i]; j := i-d;
    While (j >= Elso) And (T[j].kulcs > E.kulcs) Do Begin
      T[j+d] := T[j]; Dec(j,d)
    End{while};
    T[j+d] := E;
    Inc(i,d);
  End
End (* BeszuroRend *);

Function Valaszt(Bal,Jobb, i : Longint) : Longint;
  Var k,f,M,L:Longint; E:Elemtip;
  Begin (* Valaszt *)
    Repeat
      M:=Jobb-Bal+1; L:= M Div R;
      If M <= R Then Begin
        Beszurorend(Bal,M,1);
        Valaszt:=Bal+i-1; Break;
      End Else Begin
        For k:=1 To L Do Beszurorend(Bal+k-1,R,L);
        f:= Valaszt(Bal+R0*L, Bal+(R0+1)*L-1, L Div 2);
        f:= Feloszt(Bal,Jobb,f); {felosztás a T[f] elemmel}
        If i=f Then Begin
          Valaszt:=f; Break;
        End Else If i<f-bal Then
          jobb:=f
        Else Begin
          i:= i-(f-bal+1); bal:=f+1
        End
      End;
    Until False;
  End (* Valaszt *);
Begin LinKivalaszt := Valaszt(1,N,i) End;

```

Kiválasztás kupaccal.

A LINKIVALASZT algoritmus ugyan legrosszabb esetben is lineáris futási idejű, azonban az $O(n)$ kifejezésben a konstans elég nagy. Kis i -re gyorsabb algoritmust adhatunk kupac alkalmazásával.

```

Function KupacValaszt(Var T:Tomb;N, i:Longint) : Longint;
  Var
    E:Elemtip; k:Longint;
  Begin{KupacValaszt}
    k:=i Div 2 Downto 1 Do {kupacépítés}
      Sullyeszt(k, i);
    For k:=i+1 To N Do

```

```

{Invariáns: T[1] a tömb T[1..k] elemei közül az i-edik legkisebb}
If T[k]<T[1] Then Begin
    E:=T[1]; T[1]:=T[k]; T[k]:=E;
    Sullyeszt(1,i);
End;
KupacValaszt:=1;
End{KupacValaszt};

```

A KUPACVALASZT algoritmus futási ideje:

$$T_{lr}(n) = O(i) + (n - i)O(\lg i).$$