```
ha      c[i]    akkor
        k := 2*i
        ciklus míg k <= n
            c[k] := hamis
            k := k+i; → i-s lépésközzel megyünk
        c vége;
    h vége
    i := i+1;
c vége;
```

Kiíratás:

```
ciklus i := 2...n
  ha c[i] akkor ki : i;
c vége;
```

Példa:

Tfh: ∃ egy országban n település. Ismerjük, h. az egyik település a másikkal mekkora hosszú úttal van összekötve.

Határozzuk meg, h. melyik két település van egymáshoz a legközelebb!

Adott a távolságok sorozatai. Egészítsük ki az elemeket a települések nevével. Olyan vektor, amelynek annyi eleme van, ahány távolságot megadtunk, egy elem egy rekord, amely áll egy szám és egy név mezőből.

(Rekord típusú vektor)

Nincs ismétlés: Eger – Andornaktálya
Andornaktálya – Eger

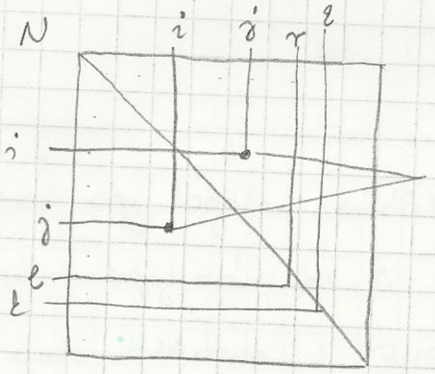
1 településpár csak 1x szerepel.

Minden pontból minden pontba él vezet. (Teljes gráf)

Ez nem más, mint a minimumkiválasztás elme.
(Ki kell választani azt, ahol a távolság minimális.)

Tfh. n db településről van szó.

Vegyünk fel n×n-es mátrixot, tudunk hivatkozni a sorszámmal.



a két település távolsága

- A k. sor k. oszlopában ∅ szerepel, mert azt mondjuk meg, h. mennyi a magától való távolság

? Mi a megoldás, Ha 2 települést nem köt össze út?

- Nem az a gond, h. mennyi a 2 azonos település távolsága, hanem, h. mi van, ha a 2 települést nem köti össze út.
- Elegendő a byte típus, arra az (r, l) pozícióba tároljuk a legnagyobb értéket. ⇒ Elegendő v. a főátló alatti v. feletti részt vizsgálni. <Elegendő egy vektorban vizsgálni (tárolni)>

Sorozatok párhuzamos feldolgozása

- Egy sorozat elemeit halmazban tároljuk. Azt a halmazt keressük, ami az uniót tárolja.
  Unióképzés az összefuttatás is. → ha tudjuk, h. a halmaz elemei rendezett sorozatban vannak eltárolva.

3. előadás

n.25.

(ábra 150)

```
begin
i:=50;
assign(f,'c:\my\!\prog_mod\ ... .txt');
rewrite(f);
while i<=n do begin
writeln (f, i,' ', o(i), ' ', p(i), ' ', c(i));
 i := i+g;
 end;
 close(f);
end;
```

o(i) ↓ osztónáma alapján megkap, h. i-ig hány prímszám van

p(i) → prímszámokkal

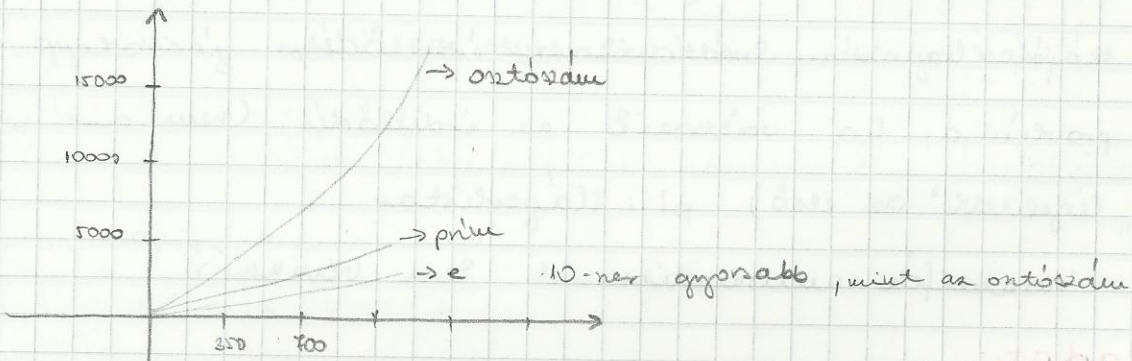
c(i) — eratosthenészi szitával

```
function e(m:word):word
var s: array [2..m] of boolean
    i,j,db:word
begin
    for j:=2 to m do s[j]:=true;
    db:=0;
    for j:=2 to round(sqrt(m)) do
      if s[j] then begin
          i:=2*j;
          while i<=m do begin
          s[i]:=false; i:=i+j; db:=db+1;
          end;
      end;
    e:=db;
end;
```

```
function o(m:word):word;
var i,j,db,os :word
begin
    db:=0;
  for j:=2 to m do begin
        os:=0
        for i:=2 to round(sqrt(j)) do begin
              if (j mod i)=0 then os:=os+1.
            db:=db+1;
            end;
        end;
        o:=db; writeln
  end,
```

14

```
function   p(n:word):word;
 var i,j,db,os :word ;
 begin
   db :=0;
 for j:=2 to n do begin
   i:=2;
 while((j mod i) <> 0) and (i<= round(sqrt(j))) do begin
   i:= i+1;  db:=db+1;
end;
end;
p:=db, writeln;
end;
```



Végrehajtási idő csökkentése – ciklusmag lefutási száma

Gyorsaság szerinti elrendezés

- Hogy megtaláljuk a sorozat egy elemét, hány önmehasonlításra van szükség.

(matematikailag nem vezethető le)

- Keressük meg egy elemet egy sorozatban

a; a sorozat nem rendezett

b; rendezett: kihasználjuk a rendezettséget

- pl.: meg van adva egy $\{a_n\}$ sorozat, és csinálunk egy $\{g_n\}$ sorozatot

tényleges adatok →

a_1	...	a_i	...	a_n
g_1	...	g_i	...	g_n

> előfordsorozat

↑ számláló

- előáll, h. melyik elemet milyen gyakorisággal használták fel. ⇒ előáll a gyakoriság
- kellő érdeklődés (keresés) után rendezzük. A g értéke ahol a legnagyobb, az lesz az első mező. ⇒ a sorozat elejére a gyakran keresett elemek kerülnek, így kb. kereséssel kevesebb számú lépéssel találjuk meg. (a ritkán keresett elemekhez sok lépés kell, de azt ritkán kell megtenni)
- ez jó, ha nem változik az érdeklődés
- probléma, ha változik az érdeklődés (nem a „legelitebb" az első) pl.: slágerlista

Valamiféle automatizmust kell bevinni!

Feladat:

Az adatsorozat ~~elemei~~ a felhasználás gyakorisága szerint van rendezve. (Előnyös a gyakran használt elemeket a sorozat elején tárolni)

A[n] összetartozó értékpárok sorozata (előfizető neve + telefonszám) Hat. meg az értékpár első elemének ismeretében a 2. elemét!

```
eljárás keres_1 ( k:egész, x: érték_1 , Y : érték_2 );
  k: = 1;
  ciklus amíg A[k].M1 ≠ x
      k: = k+1;
  c vége;
  Y: = A[k]. M2;
e vége.
```

lin kiválasztás. keresett elem helye : k.

Azonos gyakorisággal keresünk elemeket:

s: azonos gyakoriság → minden elem keresési gyakorisága

n: elemek száma (n·s keresést hajtottunk végre)

t: összehasonlítások átlagos száma

Azonos gyakoriság esetén az összehasonlítások átlagos számát keressük.

	a_1 ...	a_i ...	a_n
	s	s ...	s
összehas. száma	$1 \cdot s$	$i \cdot s$	$n \cdot s$

$$\sum_{i=1}^{n} i = \frac{n \cdot (n+1)}{2}$$

$$t = \frac{\sum_{i=1}^{n} i \cdot s}{n \cdot s} = \frac{1}{n} \sum_{i=1}^{n} i = \frac{n+1}{2}$$

1 kereséshez átlagosan ↑ összehasonlítást kell végezni.

s_i : i. elem keresési gyakorisága

(1) ! $\underbrace{s_i > s_{i+1}}$ $\forall$ i esetén

csökkenő sorrend (keresési gyakoriság szerint)

S : összes keresések száma : $\sum_{i=1}^{n} s_i$

$a_1 \; \dots \; a_i \; \dots \; a_n$

$s_1 \; > \; s_i \; \quad s_n$

$\sum_{i=1}^{n} s_i = s_1 + s_2 + s_3 + \dots + s_i + \dots + s_n$

$$t = \frac{\sum_{i=1}^{n} i \cdot s_i}{S}$$

①	②	③	④
10	8	7	5
10	2·8	3·7	4·5

összehas. → keresés

Bizonyítandó : t minimális $s_i > s_{i+1}$ ($\forall i$) esetén

ha $i < j$, $s_i > s_j$

mivel a keresési szám a gyakoriság szerint csökk. sorrendben van, így az i összehasonlítások száma kisebb, mint j

i esetén s_j

j esetén s_i

10 8 7 5

⑦ ⟷ ⑩ megcseréljük

Felkikeresgálások száma közötti különbség :

$$(\underbrace{i \cdot s_i + j \cdot s_j}_{\text{kisebb}}) - (\underbrace{j \cdot s_i + i \cdot s_j}_{\text{nagyobb}}) = i(s_i - s_j) + j(s_j - s_i) < 0$$

pozitívak

+ −

$|s_i - s_j| = |s_j - s_i|$

Pl.: ! rendre $a_i = n-i+1$ $(\forall i)$ esetén

az első elemet n-szer } keressük
az utolsót 1-szer

Az összes keresések száma: $\sum_{i=1}^{n}(n-i+1) = \frac{n(n+1)}{2}$

pontos, tehát $\frac{1}{x}$-szeresekről.

$$t = \frac{2}{n(n+1)} \cdot \sum_{i=1}^{n} i(n-i+1) = \frac{2}{n(n+1)}\left[\sum_{i=1}^{n} n\cdot i - \sum_{i=1}^{n} i^2 + \sum_{i=1}^{n} i\right] =$$

összehasonlítások száma átlagosan

$$\left\langle \sum i(n-i+1) = \sum(in - i^2 + i) \right\rangle$$

$$= \frac{2}{n(n+1)}\left[n\frac{n(n+1)}{2} - \frac{n\cdot(n+1)(2n+1)}{2\cdot 3} + \frac{n(n+1)}{2}\right] =$$

$$\left\langle \sum_{i=1}^{n} in = 1n + 2n + 3n + \ldots \right.$$
$$\left. n(1+2+3+\ldots) \right\rangle$$

$$= n+1\frac{(2n+1)}{3} = \underline{\underline{\frac{n+2}{3}}}$$

- Gyakoriság szerinti elrendezés jó!
Ha minden elemet azonos gyakorisággal keresünk $\Rightarrow$
így is több az összehasonlítások száma.

- Megváltozhat egyes elemekre vonatkozó a keresési gyakoriság
- Ha rendezzük a keresési gyakoriságot $\Rightarrow$
ha sűrűn: sok a gépidő
ha ritkán: a legnépszerűbb lehet az utolsó helyen $\Rightarrow$
lassú.

- Ha megtaláltunk egy elemet ⇒ 1-gyel nő a keresési gyakoriság ⇒ cseréljük meg az előtte lévővel az elemet ⇒ a gyakoriak a sorozat elejére vándorolnak.

```
eljárás keres_2 (k: egész, X: elter_1, Y: elter_2);
    k := 1;
    ciklus míg A[k].M1 ≠ X
        k := k+1;
    c vége;
    Y := A[k].M2;
    Ha k>1 akkor Csere (A[k], A[k-1]);
e vége;
```

$\left\langle \sum_{i=1}^{n} i \cdot s_i = S \right.$

$S - (i s_i + j s_j) = S'$

$i s_j + j s_i + S'$

$i s_i + j s_j + S$

a többi elemre nem változott meg az összehasonlítások száma, ezért ezzel nem kell foglalkozni. $\rangle$

Végrehajtási idő csökkentése - ciklusmag lefutásának száma

Sorozat elemeinek csoportos feldolgozása

Más tulajdonság alapján is kereshetünk bizonyos elemek egymáshoz közel.

Feladat:

Az ásványkincs-készlet becslése céljából kutatófúrásokat végeztek. Minden fúráskor többször (adott mélységet elérve) mintát vettek és feljegyezték a mintában található (l) ásványi anyag mennyiségét, összesen (n)-szer. Adjuk meg (k) fúrás adatait!

- 1 mérés ⇒ 1 mintavételt jelent
- minden mérés jellemezhető:
  - hányas fúrás

    és nem egyértelműen adja meg (1 fúrásnál több mintavétel.)
  - mélység
  - ásványi adatok mennyisége

1 mérés $\overset{!!}{(l+2)}$ adattal jellemezhető

```
eljárás Kiválogat_1(n, k: egész);
    ciklus i:=1..n;
        ha A[i].s = k akkor Ki: A[i].mérés
    c vége;
e vége
```

↓ k. fúrás összes mintavételének megtalálását segíti elő.

Specialitás:

Azonos sorszámú fúrások mérései folyamatosan helyezkednek el a sorozatban.