



Véletlen tesztek

N elem keverése

Véletlen keverés:

- Az 1 és N közötti számokra minden lépésben cseréljük meg az i . elemet az $i..N$ intervallum egy véletlenszerű elemével!

```
Ciklus i=1-től N-ig
```

```
  x[i]:=i
```

```
Ciklus vége
```

```
Ciklus i=1-től N-ig
```

```
  j:=véletlen(i..N)
```

```
  csere(x[i],x[j])
```

```
Ciklus vége
```

Be kell látni, hogy így minden elem egyenlő eséllyel kerül bármely helyre!

Működik tetszőleges N darab elemre is, ők legyenek kezdetben az x vektorban!

N elem keverése

Véletlen keverés:

- Generáljunk N darab 0 és $N-1$ közötti számot lineáris kongruencia módszerrel, megfelelő $x[0]$, a , b értékekkel!

```
Ciklus i=1-től N-ig  
  x[i] := (x[i-1]*a+b) mod N  
Ciklus vége
```

Feltétel: N és b relatív prím, $a-1$ -et osztja N minden prímosztója, és ha 4 osztója N -nek, akkor $a-1$ -nek is.

Így adott $x[0]$, a , b értékekre nyelvfüggetlen megismételhető tesztorozatot kapunk.

Zempléni András: <https://zempleni.elte.hu/rnd17.pdf>

M elemből N kiválasztása

Próbák és tévedések módszere:

- Tároljuk, hogy mely 1 és M közötti értékek szerepeltek már a generálásban, és ha ilyet választottunk, akkor válasszunk újra – nagy M-re érdemes, kicsire lassú!

```
volt := (hamis, ..., hamis)
```

```
Ciklus i=1-től N-ig
```

```
  Ciklus
```

```
    x[i] := véletlen(1..M)
```

```
    amíg volt[x[i]]
```

```
  Ciklus vége
```

```
    volt[x[i]] := igaz
```

```
Ciklus vége
```

M elemből N kiválasztása

Véletlen kiválogatás:

- Bármelyik N/M valószínűséggel kerüljön be a kiválasztottak közé!

$$Kiválaszt(M, N) = \begin{cases} \text{első kell és } Kiválaszt(M-1, N-1) & \text{ha véletlenszám} < N/M \\ \text{első nem kell és } Kiválaszt(M-1, N) & \text{egyébként} \end{cases}$$

```
Kell:=N; Van:=M; db:=0
```

```
Ciklus i=1-től M-ig
```

```
  Ha véletlenszám<Kell/Van
```

```
    akkor db:=db+1; x[db]:=i; Kell:=Kell-1
```

```
  Van:=Van-1
```

```
Ciklus vége
```

Be kell látni, hogy így pontosan N elemet választunk ki!

M elemből N kiválasztása, legfeljebb K ismétlődéssel



Véletlen kiválogatás:

- Bármelyik $N/M \cdot K$ valószínűséggel kerüljön be a kiválasztottak közé, de K -szor nézzünk minden elemet!

```
Kell:=N; Van:=M*K; db:=0
```

```
Ciklus i=1-től M-ig
```

```
  Ciklus j=1-től K-ig
```

```
    Ha véletlenszám<Kell/Van
```

```
      akkor db:=db+1; x[db]:=i; Kell:=Kell-1
```

```
    Van:=Van-1
```

```
  Ciklus vége
```

```
Ciklus vége
```

Sok elemből N kiválasztása



Véletlen kiválogatás:

- Az új véletlen elemre nézzük meg, hogy már szerepelt-e a kiválogatottak között – M sokkal nagyobb, mint N !

```
Ciklus i=1-től N-ig
```

```
  Ciklus
```

```
    j:=véletlen(1..M)
```

```
    k:=1
```

```
    Ciklus amíg k<i és x[k]≠j
```

```
      k:=k+1
```

```
    Ciklus vége
```

```
  amíg k<i
```

```
  Ciklus vége
```

```
  x[i]:=j
```

```
Ciklus vége
```

Véletlen monoton sorozatok

Véletlen növelés:

- Ha a számok értékeire nincs felső korlát, akkor nagyon egyszerű véletlen monoton növekvő sorozatot gyártani, ahol az értékek legfeljebb K -val nőnek:

```
x[1]:=véletlen(1..K)
```

```
Ciklus i=2-től N-ig
```

```
    x[i]:=x[i-1]+véletlen(1..K)
```

```
Ciklus vége
```


Véletlen monoton sorozatok

Véletlen kiválogatás:

- Ha a számok értékeire van felső korlát (M), akkor másképp kell hozzáállni, N darab intervallum határt kell generálnunk, az utolsó így mindig legfeljebb M lesz:

```
Kell:=N; Van:=M; db:=0
```

```
Ciklus i=1-től M-ig
```

```
  Ha véletlenszám<Kell/Van
```

```
    akkor db:=db+1; x[db]:=i; Kell:=Kell-1
```

```
  Van:=Van-1
```

```
Ciklus vége
```

Sorozatok véletlen összefésülése



Sorozatok összefésülése:

- Azt tároljuk, hogy hány elemet kell még az első, illetve a második sorozatból az eredménybe tenni.

```
i:=1; j:=1; k:=0; d:=N; e:=M
```

```
Ciklus amíg d+e>0
```

```
  k:=k+1
```

```
  Ha véletlenszám<d/(d+e)
```

```
    akkor z[k]:=x[i]; i:=i+1; d:=d-1
```

```
  különben z[k]:=y[j]; j:=j+1; e:=e-1
```

```
Ciklus vége
```

Véletlen intervallumok

Diszjunkt intervallumok:

- Olyan sorozatokat generálunk, amelyek kétféle típusú elem-ből állnak (igaz, hamis), az igaz értékűek az intervallumok elemei (ha N nem túl nagy):

Ciklus $i=1$ -től N -ig

Ha véletlenszám $< P$ akkor $x[i] := \text{igaz}$

különben $x[i] := \text{hamis}$

Ciklus vége

$db := 0$; $x[0] := \text{hamis}$; $x[n+1] := \text{hamis}$

Ciklus $i=1$ -től N -ig

Ha $x[i]$ és nem $x[i-1]$ akkor $db := db + 1$; $k[db] := i$

Ha $x[i]$ és nem $x[i+1]$ akkor $v[db] := i$

Ciklus vége

Véletlen intervallumok

Diszjunkt intervallumok:

- Olyan sorozatokat generálunk, amelyek kétféle típusú elem-ből állnak (igaz, hamis), az igaz értékűek az intervallumok kezdő és végpontjai (legfeljebb K hosszúak):

```
log:=igaz; i:=véletlen(K); db:=0; x:=(hamis,...,hamis)
```

```
Ciklus amíg  $i \leq N$ 
```

```
  x[i]:=log; i:=i+véletlen(K)
```

```
Ciklus vége
```

```
Ciklus  $i=1$ -től  $N$ -ig
```

```
  Ha x[i] akkor
```

```
    Ha log akkor db:=db+1; k[db]:=i; log:=hamis
```

```
    különben v[db]:=i; log:=igaz
```

```
Ciklus vége
```


Véletlen intervallumok

Diszjunkt intervallumok:

- N darab kezdetet és véget generálunk (az intervallumok legfeljebb K hosszúak, a távolságuk legfeljebb L):

```
i:=1; kezd[i]:=véletlen(K);
```

```
Ciklus amíg  $i \leq N$ 
```

```
    vég[i]:=kezd[i]+véletlen(K)
```

```
    i:=i+1; kezd[i]:=vég[i-1]+véletlen(L)
```

```
Ciklus vége
```

Véletlen intervallumok

Nem diszjunkt intervallumok:

- Egyenlő eséllyel generáljuk bármely részintervallumot ($E[i] < U[i]$):

```
Ciklus i=1-től M-ig
```

```
  E[i] := Véletlen(1..N)
```

```
  Ciklus
```

```
    U[i] := Véletlen(1..N)
```

```
    amíg E[i] = U[i]
```

```
  Ciklus vége
```

```
  Ha E[i] > U[i] akkor Csere(E[i], U[i])
```

```
Ciklus vége
```

Véletlen intervallumok

Legfeljebb K hosszú intervallumok:

- Egyenlő eséllyel generáljuk az intervallumok kezdőpontját:

Ciklus $i=1$ -től M -ig

$E[i] := \text{Véletlen}(1..N)$

$U[i] := \text{Véletlen}(E[i], \min(N, E[i] + K - 1))$

Ciklus vége

- Így nem egyenlő esélyűek lesznek az intervallumok. Ha minden hosszúságúból egyenlő eséllyel kell generálni, akkor:

Ciklus $i=1$ -től M -ig

$h := \text{véletlen}(1..K)$

$E[i] := \text{Véletlen}(1..N-h+1); U[i] := E[i] + h - 1$

Ciklus vége

- Még az lehet gond, hogy a különböző hosszú intervallumokból különböző darabszámú van.

Véletlen intervallumok

Legfeljebb K hosszú biztosan átfedő intervallumok:

- Egyenlő eséllyel generáljuk az intervallumok kezdőpontját, az előző intervallum belsejében (a vége is lehet belül):

$E[1] := \text{Véletlen}(1..N)$

$U[1] := \text{Véletlen}(E[1], \min(N, E[1] + K - 1))$

Ciklus $i=2$ -től M -ig

$E[i] := \text{Véletlen}(E[i-1]..U[i-1])$

$U[i] := \text{Véletlen}(E[i], \min(N, E[i] + K - 1))$

Ciklus vége

Véletlen fa

Legyen egy N elemű fa gyökere az 1-es pont! Minden új pontot kössünk valamely korábbi ponthoz! N elemű fának $N-1$ éle van.

```
Ciklus i=2-től N-ig
    él[i-1,1]:=i; él[i-1,2]:=véletlen(1..i-1)
Ciklus vége
```

Véletlen bináris fánál nézni kell, hogy egy pontból ne induljon ki 2-nél több él.

```
db[] := (0, ... 0)
Ciklus i=2-től N-ig
    Ciklus
        x:=véletlen(1..i-1)
        amíg db[x]=2
    Ciklus vége
    él[i-1,1]:=i; él[i-1,2]:=x; db[x]:=db[x]+1
Ciklus vége
```

Véletlen gráf

N csomópontból álló irányítatlan gráfba véletlen M darab él:

```
van:=N*(N-1) div 2; kell:=M; db:=0
```

```
Ciklus i=1-től N-1-ig
```

```
  Ciklus j=1-től N-1-ig
```

```
    Ha véletlemszám<kell/van akkor
```

```
      akkor db:=db+1; él[db,1]:=i; él[db,2]:=j
```

```
      kell:=kell-1
```

```
    van:=van-1
```

```
  Ciklus vége
```

```
Ciklus vége
```

Véletlen tesztek

Amiről szó volt:

- Keverés
- Véletlen elemek kiválogatása
- Véletlen sorozatok
- Véletlen intervallumok
- Véletlen fák és gráfok