Stuktogramok

1. Edényrendezés:

bucket_sort($A[1..n]$)

Hozzuk létre a $B[0..n-1]$ edényeket
$j := 0..n-1$
Legyen $B[j]$ üres lista
$i := 1..n$
Szúrjuk be az $A[i]$ elemet a $B[\lfloor n * (A[i].kulcs) \rfloor]$ listába
$j := 0..n-1$
Rendezzük monoton növekvően a $B[j]$ listát
Fűzzük össze sorban a $B[0]$, $B[1]$, $\ldots$, $B[n-1]$ listákat az L listába
Másoljuk vissza az L lista tartalmát sorban az $A[1..n]$ tömbbe

műveletigény: általában: o(n), de lehet o(n^2) is.

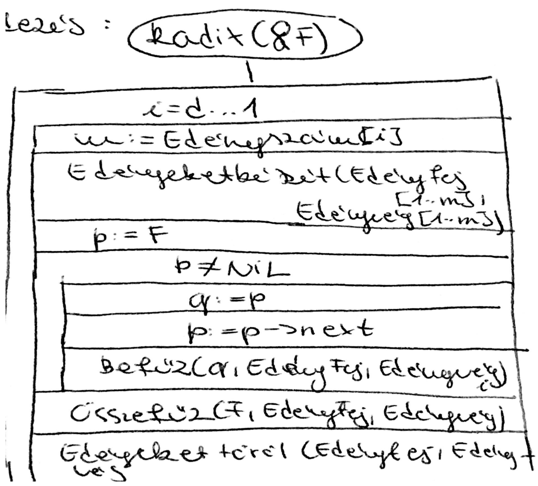
2. Leszámoló rendezés:

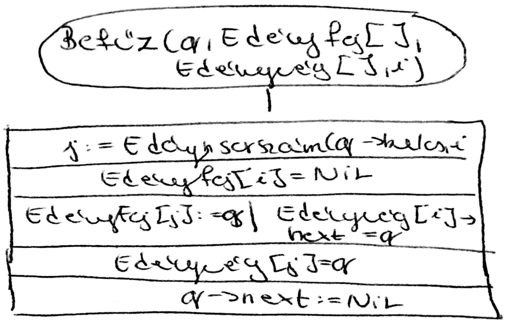
counting_sort($A[1..n], k, \varphi, B[1..n]$)

Hozzuk létre a $C[0..k]$:$\mathbb{N}$ tömböt
$j := 0..k$
$C[j] := 0$
$i := 1..n$
$C[\varphi(A[i])]++$
$j := 1..k$
$C[j] += C[j-1]$
$i := n..1 \quad (-1)$
$j := \varphi(A[i])$
$B[C[j]] := A[i]$
$C[j]--$

műveletigény: $\Theta(n + k)$ ($k \in O(n)$, $\Theta(n + k) = \Theta(n)$, azaz $T(n) \in \Theta(n)$).

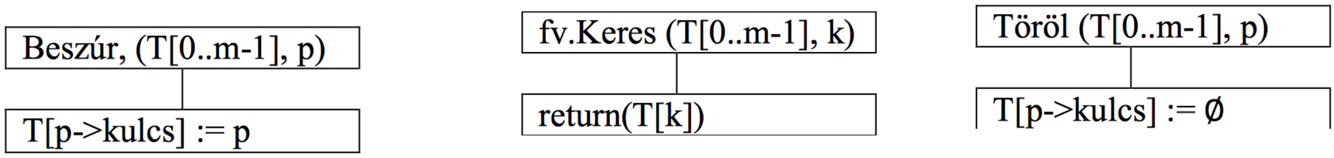
3. Radix rendezés – Edényrendezéssel: műveletigény: $\Theta(n)$



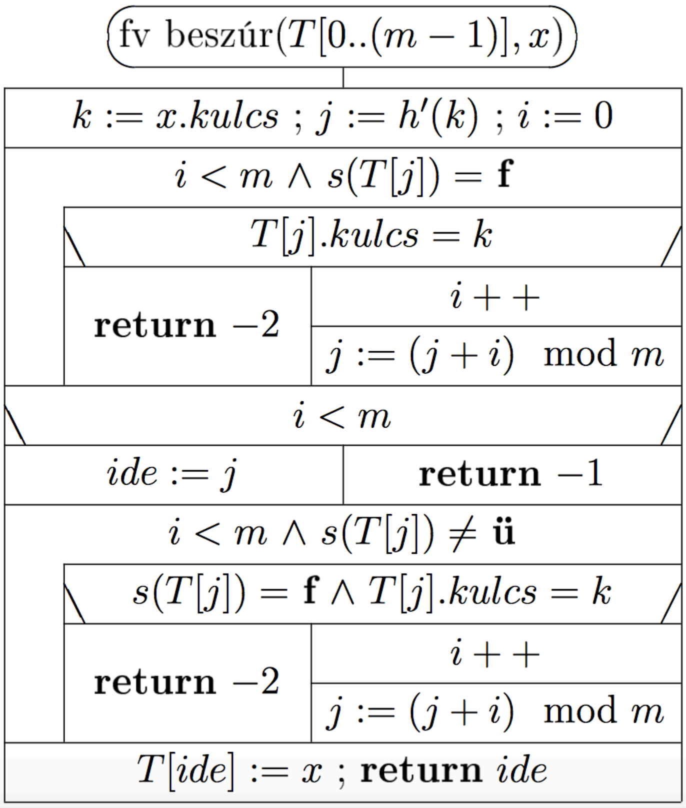


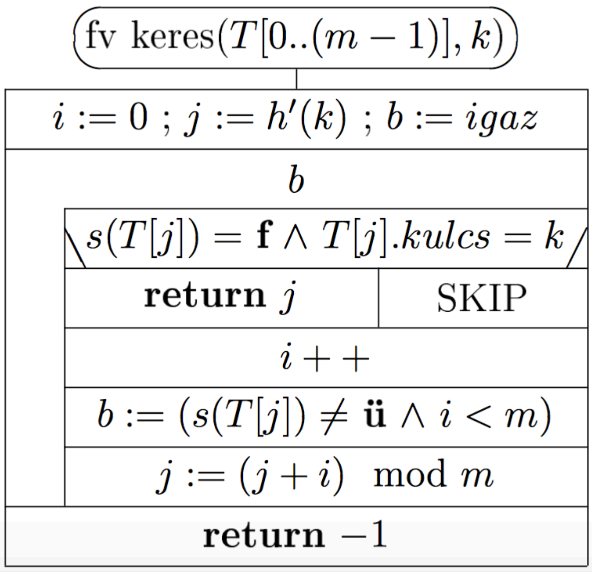
4. Hashelés:

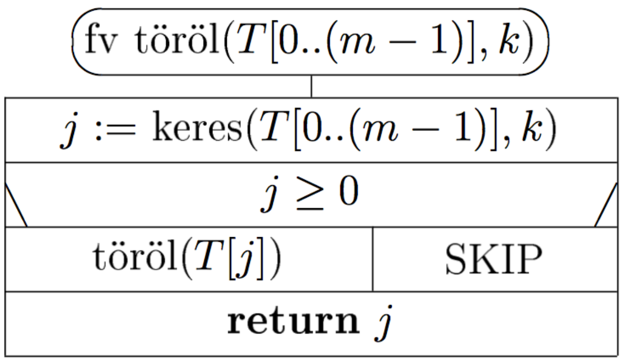
a. Direkt címzés-műveletek:

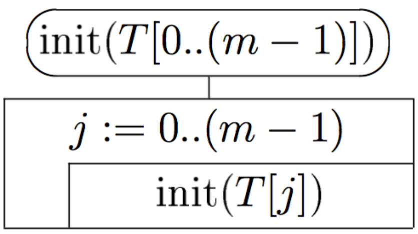


b. Nyílt címzés-műveletek:



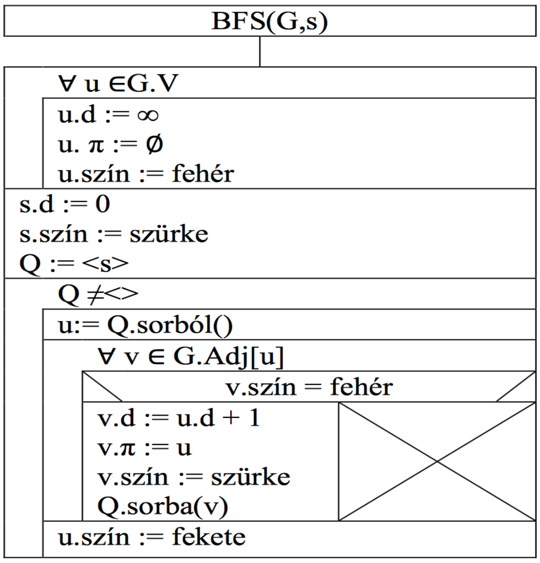






Gráfok

5. Szélességi bejárás:



műveletigény:

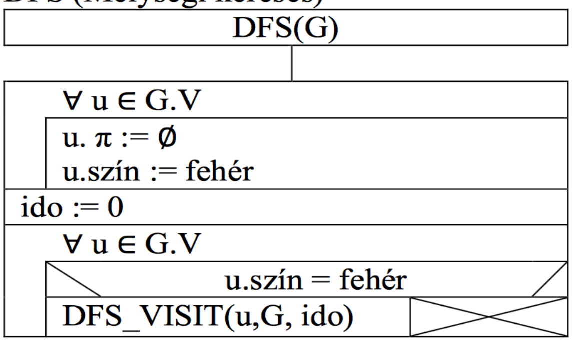
$T(n,e) \in O(n+e)$ BFS

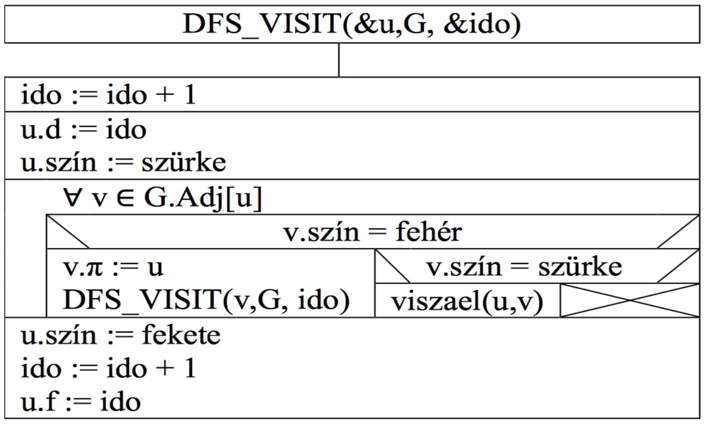
$MT(n,e) \in \theta(n+e)$ BFS

$mT(n,e) \in \theta(n)$

6. Mélységi bejárás:

DFS (Mélységi keresés)



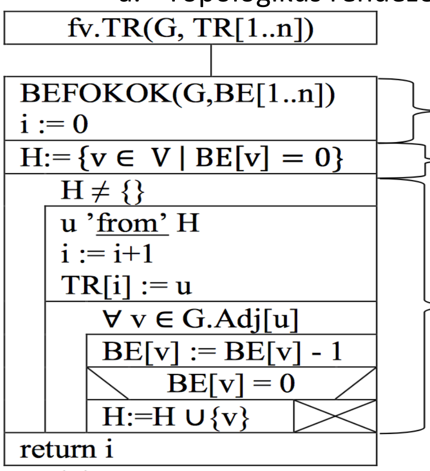


u.d = elérési idő (discovery time)

u.f = befejezési idő (finishing time)

műveletigény: $T(n,e) \in \theta(n+e)$

7. Mélységi bejárás alkalmazásai:
    a. Topologikus rendezés:



$\Theta(n+e)$

$\Theta(n)$

'from' ← kivesz H-ból egy elemet

a kör csúcsai nem kerülnek H-ba, így az eredménybe (TR[1..n]) sem.

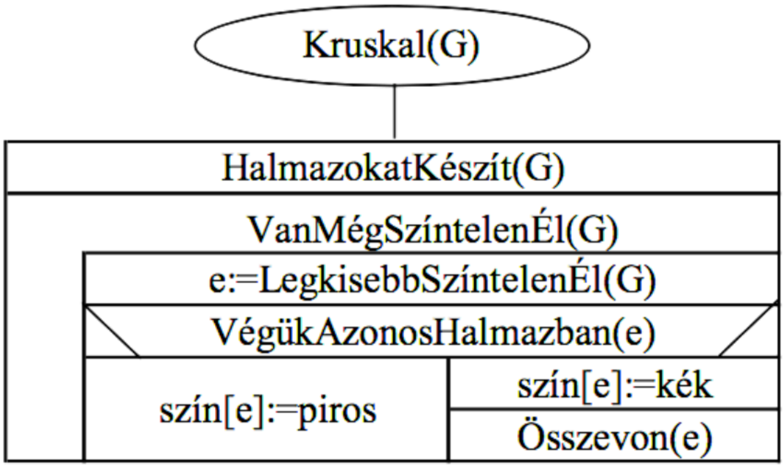
$O(n+e)$

ha i=n, minden csúcsot feldolgoztunk

$T_{TR}(n) \in O(n+e)$.

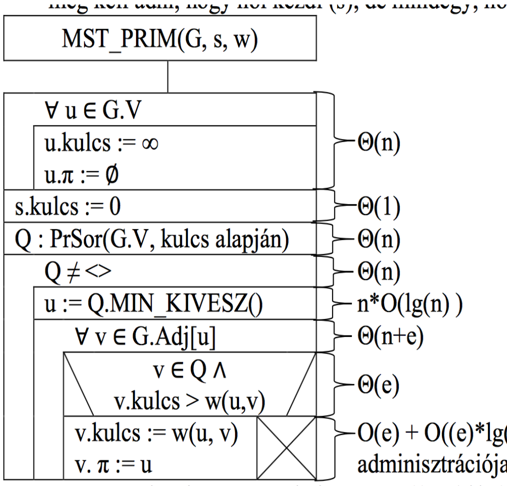
8. Szélességi bejárás alkalmazásai:

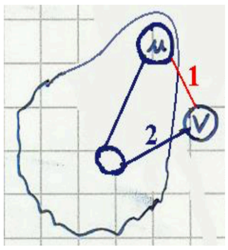
a. Kruskal



műveletigény: $O(e \log n)$

b. Prím

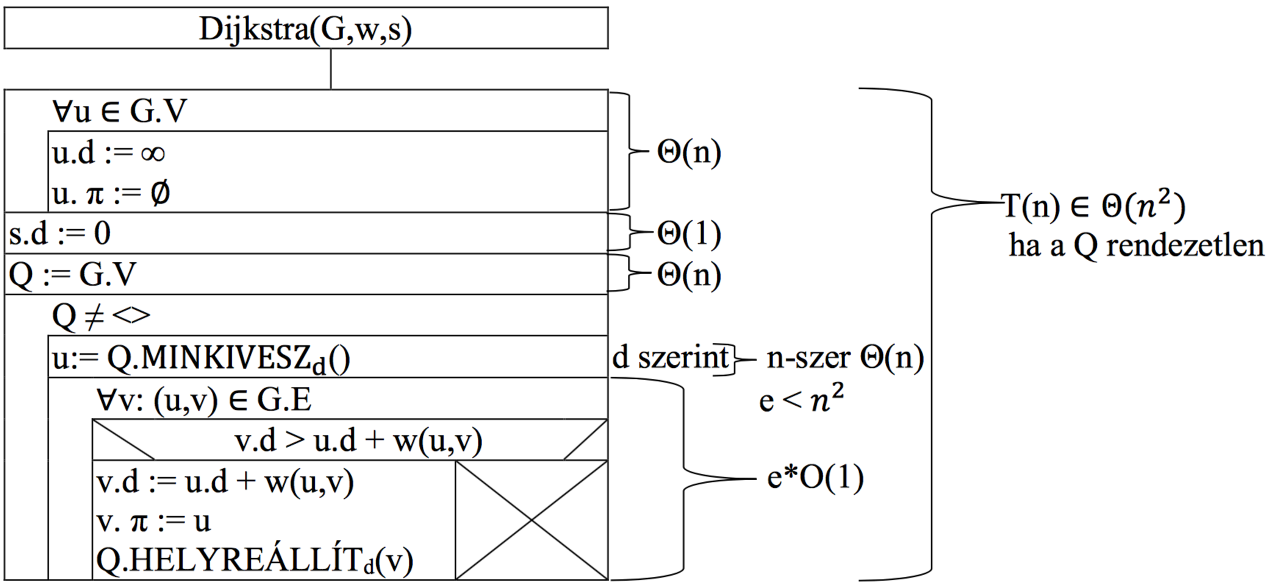




Lehet, hogy az u hozzávétele után v közelebb kerül a részleges feszítőfához.

Az egész algoritmus műveletigénye: O((n+e)*lg(n)), de a gráf összefüggő (e ≥ n-1), azaz O(e*lg(n))

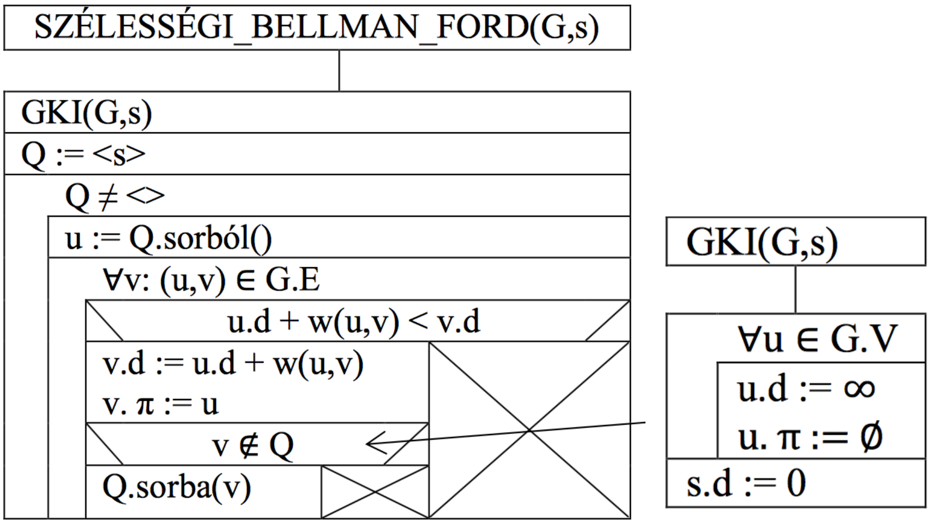
c. Dijkstra



műveletigény:

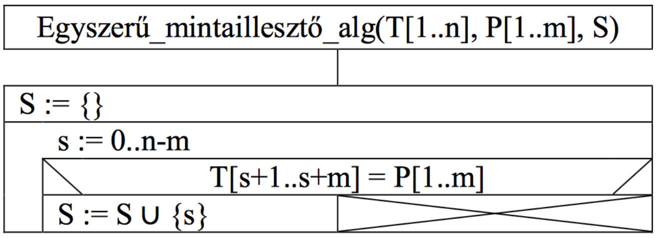
- ritka gráfra O(n*(n+e)*log n) = O(n^2*log n)
- sűrű gráfra O(n^3*log n), nem jó választás…
  - ehelyett a vektorpáros Dijkstra stabil O(n^3)-öt (n * O(n^2)) fut.

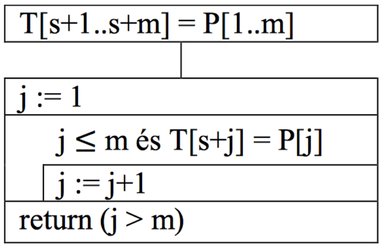
d. Bellman-Ford:



Mintaillesztés

9. Brute-Force





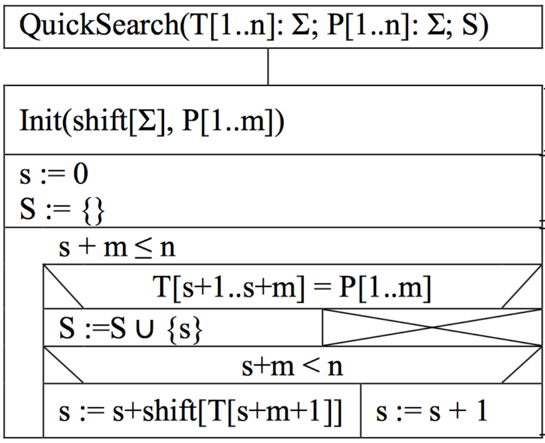
O(m)
MT(m) ∈ Θ(m)
mT(m) ∈ Θ(1)

Az egész eljárás műveletigénye:

MT(n,m) ∈ Θ((n-m+1)*m) = Θ((n-m)*m)

mT(n,m) ∈ Θ(n-m+1) = Θ(n-m) n > m esetén

10. Quick Search



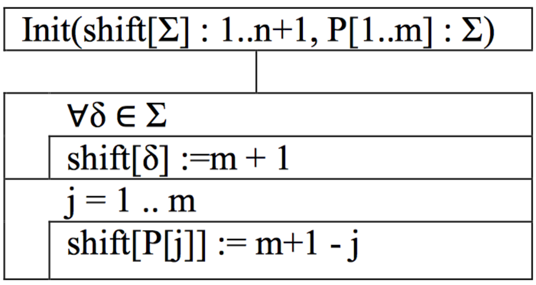
d konstans

$\Theta(m + \overbrace{|\Sigma|}^{d}) = \theta(m+d) = \Theta(m)$

$\Theta(1)$

$mT(n,m) \in \Theta\left(\frac{n}{m+1}\right)$ ← pl: T és P diszjunkt

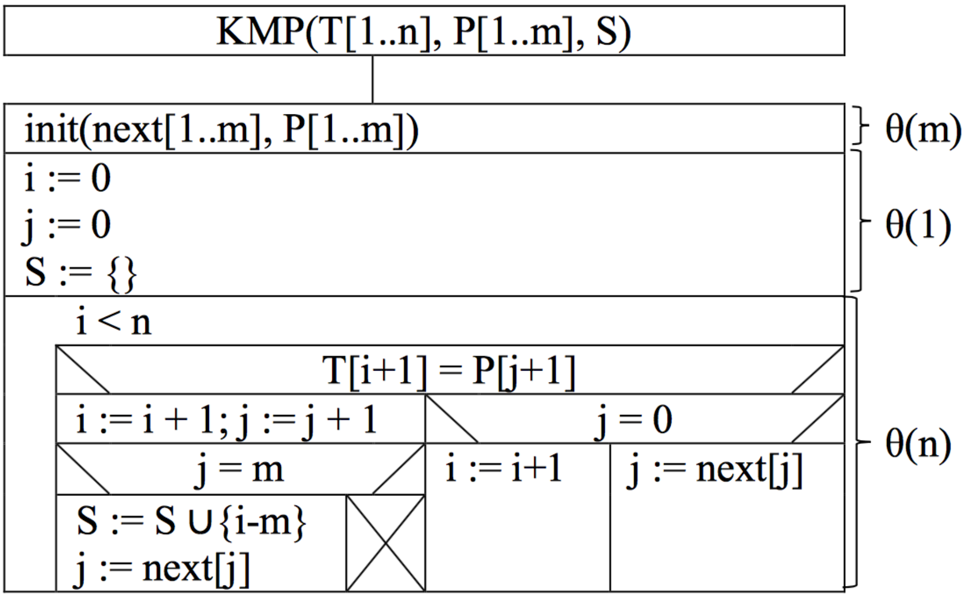
MT(n,m) ∈ Θ((n-m+1)*m) ← pl: T= A…A és p= A…A



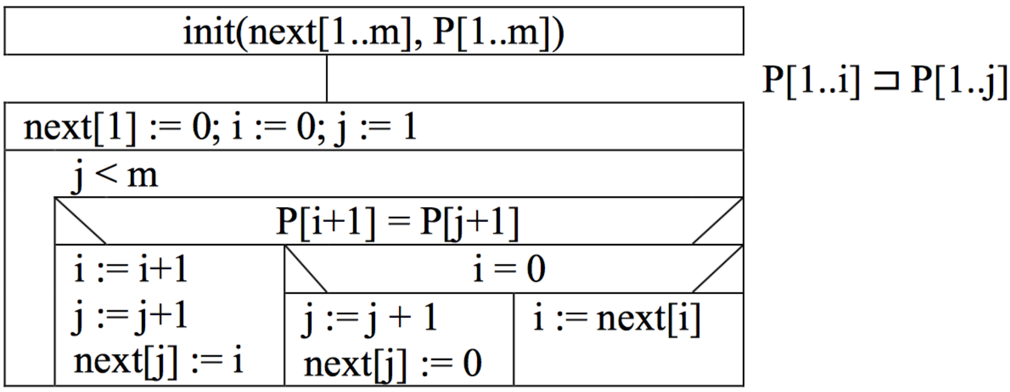
$\Theta(d)$

$\Theta(m)$

11. KMP



műveletigény: o(n)



Tömörités

12. LZW