

# Algoritmusok és adatszerkezetek I.

Gyakorlat – 2019.04.15.

Oszd meg és uralkodj stratégia

# Oszd meg és uralkodj! - stratégia

- Több részfeladatra bontás, amelyek hasonlóan oldhatók meg, lépései:
  - A triviális eset
    - amikor nincs rekurzív hívás
  - Felosztás
    - megadjuk a részfeladatokat, amikre a feladat lebontható
  - Uralkodás
    - rekurzívan megoldjuk az egyes részfeladatokat
  - Összevonás
    - az egyes részfeladatok megoldásából előállítjuk az eredeti feladat megoldását

# Oszd meg és uralkodj! - stratégia

- Ezek alapján a következőképpen fogunk gondolkodni:
  - Mi a leállás (triviális eset) feltétele? Hogyan oldható meg ilyenkor a feladat?
  - Mi az általános feladat alakja? Mik a paraméterei? Ebből kapjuk meg a rekurzív eljárásunk specifikációját.
  - Milyen paraméterértékekre kapjuk a konkrét feladatot? Ezekre fogjuk meghívni kezdetben az eljárást!

# Oszd meg és uralkodj! - stratégia

- Ezek alapján a következőképpen fogunk gondolkodni:
  - Hogyan vezethető vissza a feladat hasonló, de egyszerűbb részfeladatokra? Hány részfeladatra vezethető vissza?
  - Melyek ilyenkor az általános feladat részfeladatainak a paraméterei? Ezekkel kell majd meghívni a rekurzív eljárást!
  - Hogyan építhető fel a részfeladatok megoldásaiból az általános feladat megoldása?

# Oszd meg és uralkodj! - stratégia

## Gyorsrendezés (quicksort):

- megosztás:  $X_1, \dots, X_{k-1} \mid X_k \mid X_{k+1}, \dots, X_n$  szétválogatás ahol  
 $i, j (1 \leq i \leq k; k \leq j \leq n): X_i \leq X_j$
- uralkodás: mindkét részt ugyanazzal a módszerrel felbontjuk két részre, rekurzívan
- összevonás: automatikusan történik a helyben szétválogatás miatt
- triviális eset:  $n \leq 1$

# Oszd meg és uralkodj! - stratégia

Összefésüléssel rendezés (mergesort):

- megosztás: a sorozat két részsorozatra bontása (középen)

$$\boxed{X_1, \dots, X_k} \quad \boxed{X_{k+1}, \dots, X_n}$$

- uralkodás: a két sorozat rendezése (rekurzívan)
- összevonás: a két rendezett részsorozat összefésülése
- triviális eset:  $n \leq 1$

# Összefésül (két rendezettből egy rendezett)

**Függvény Összefésül (A,B) :**

ia:=1; ib:=1; Nc:=0;

**Ciklus amíg** ia<=Na vagy ib<=Nb

**Ha** (ia<=Na) és (((ib<=Nb) és (A[ia]<B[ib])) vagy (ib>Nb)) **akkor**

Nc:=Nc+1; C[Nc]:=A[ia]; ia:=ia+1;

**Különben**

Nc:=Nc+1; C[Nc]:=B[ib]; ib:=ib+1;

**Elágazás vége**

**Ciklus vége**

Összefésül:=C;

**Eljárás vége.**

# Összefésüléssel rendezés

**Függvény Rendez (A) :**

**Ha** Hossz (A) > 1 **akkor**

K := Hossz (A) / 2 ;

Rendez := Összefésül (Rendez (A [ :K ] ) , Rendez (A [ K+1 : ] ) ) ;

**Elágazás vége**

**Függvény vége ;**



# Szétválogatás

**Eljárás Szétválogatás (A, E, V, K) :**

segéd := A[E]

**Ciklus amíg**  $E < V$

**Ciklus amíg** ( $E < V$  és  $A[V] > \text{segéd}$ )

$V = V - 1$

**Ciklus vége**

**Ha**  $E < V$  **akkor**

$A[E] := A[V]; E := E + 1$

**Ciklus amíg** ( $E < V$  és  $A[E] < \text{segéd}$ )

$E := E + 1$

**Ciklus vége**

**Ha**  $E < V$  **akkor**  $A[V] := A[E]; V := V - 1;$

**Elágazás vége**

**Ciklus vége**

$A[E] := \text{segéd}; K = E$

**Eljárás vége.**

# Quicksort (gyorsrendezés)

**Eljárás Quick (A, E, V) :**

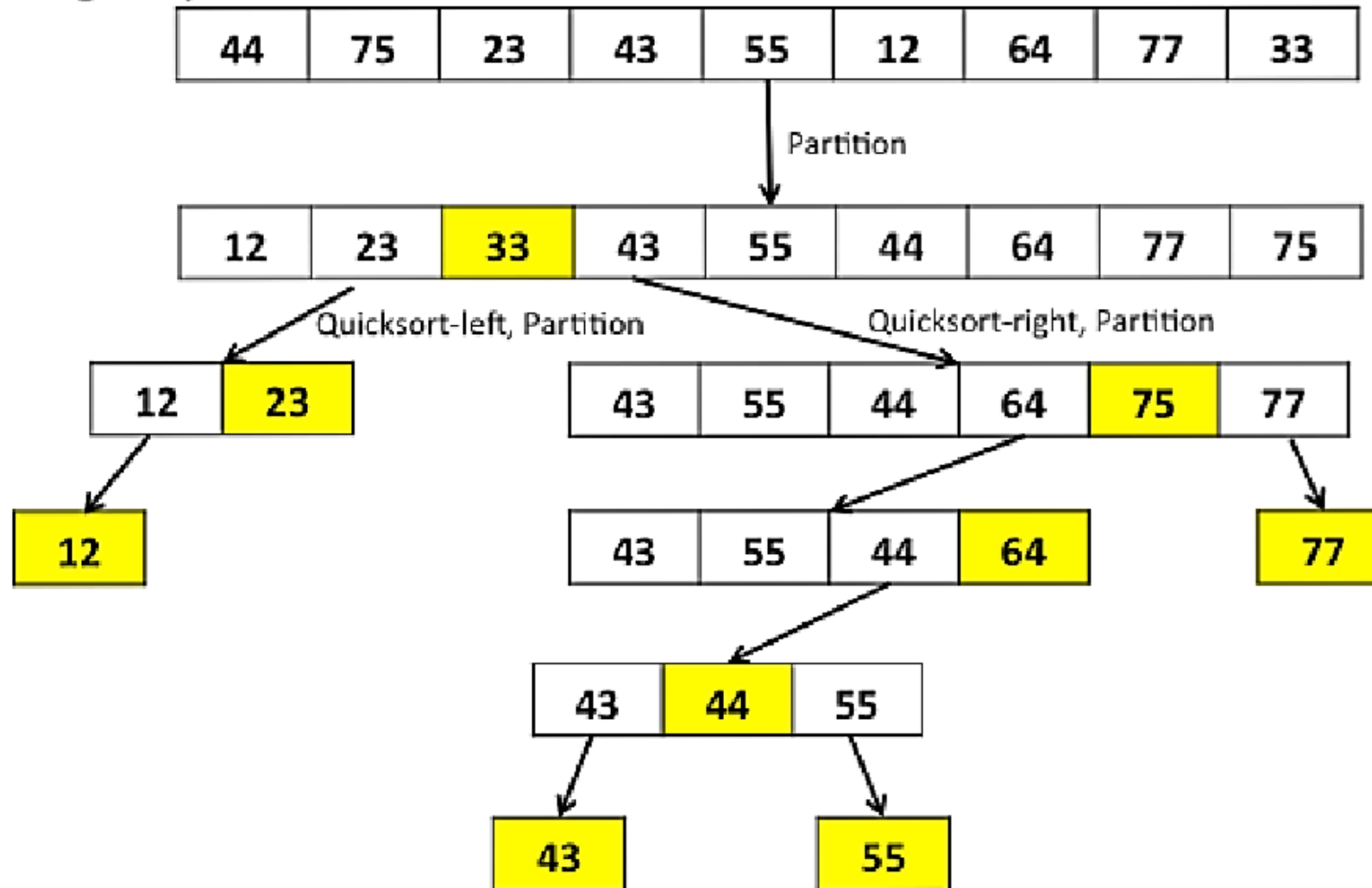
Szétválogat (A, E, V, K)

**Ha**  $K - E > 1$  **akkor** Quick (A, E, K-1) ;

**Ha**  $V - K > 1$  **akkor** Quick (A, K+1, V) ;

**Eljárás vége ;**

Starting array



Resulting array



\* Ez a példa az első elem helyett az utolsót veszi referenciának.