



Tömb

$A : \overset{L}{B}[n]$

$B : \text{bit}[n]$

$C : \mathbb{N}[n]$

$c[0]$	$c[1]$	...	$c[n-1]$
--------	--------	-----	----------

$C.\text{length} = n$

$A/1 : \mathbb{Z}[n]$

$A[1]$	...	$A[n]$
--------	-----	--------

---

$A : \mathbb{T}[n]$

↑  
abstrakt elem típus

1. változat

$$y = a_0 + a_1 * x + a_2 * x * x + a_3 * x * x * x + \dots$$

Polinom1( $A: \mathbb{R}[]$ ;  $x: \mathbb{R}$ ):  $\mathbb{R}$

$y := A[0]$	1
$i = 1 \text{ to } A.length - 1$	$n + 1$
$h := 1$	$n$
$j = 1 \text{ to } i$	$2 + 3 + \dots + n + 1$
$h := h * x$	$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$
$y := y + A[i] * h$	$n$
return $y$	1

hányszor  
hajtjuk végre?

$$A.length = n + 1$$

összeadások száma  $O \in \Theta(n)$

szorzások száma  $Sz \in \Theta(n^2)$

$$\Theta(n^2)$$

eljárás hívások  
száma

+  
ciklus iterációk  
száma

$$= 1 + n + \frac{n(n+1)}{2} \in \Theta(n^2)$$



2. változat

Polinom 2( $A: \mathbb{R}[]; x: \mathbb{R}$ ):  $\mathbb{R}$

$y := A[0]$

$h := 1$

$i = 1$  to  $A.length - 1$

$h := h * x$

$y := y + A[i] * h$

return  $y$

1

1

$n+1$

$n$

$n$

1

$$O = n \in \Theta(n)$$

$$S = 2n \in \Theta(n)$$

$$T = n+1 \in \Theta(n)$$

### 3. valtorat

$$P(x) = \left( \dots \left( \left( a_n * x + a_{n-1} \right) * x + a_{n-2} \right) * x + a_{n-3} \dots \right) * x + a_0$$

Polinom 3( $A: \mathbb{R}[]; x: \mathbb{R}$ ):  $\mathbb{R}$

$y := A[A.length - 1]$	1
$i = A.length - 2$ downto 0	$n + 1$
$y := y * x + A[i]$	$n$
return $y$	1

$$\ddot{O} = n \in \Theta(n)$$

$$S = n \in \Theta(n)$$

$$T = 1 + n \in \Theta(n)$$

# Renderési algoritmusok

input: elemek egy tömbben

output: átrendezett nem-csökkenő sorrendben

Beszúrásos rendezés Insertion Sort

Kiválasztásos rendezések: Max Kiv-Resz, (Min) Max Selection Sort

Buborék rendezés Bubble Sort

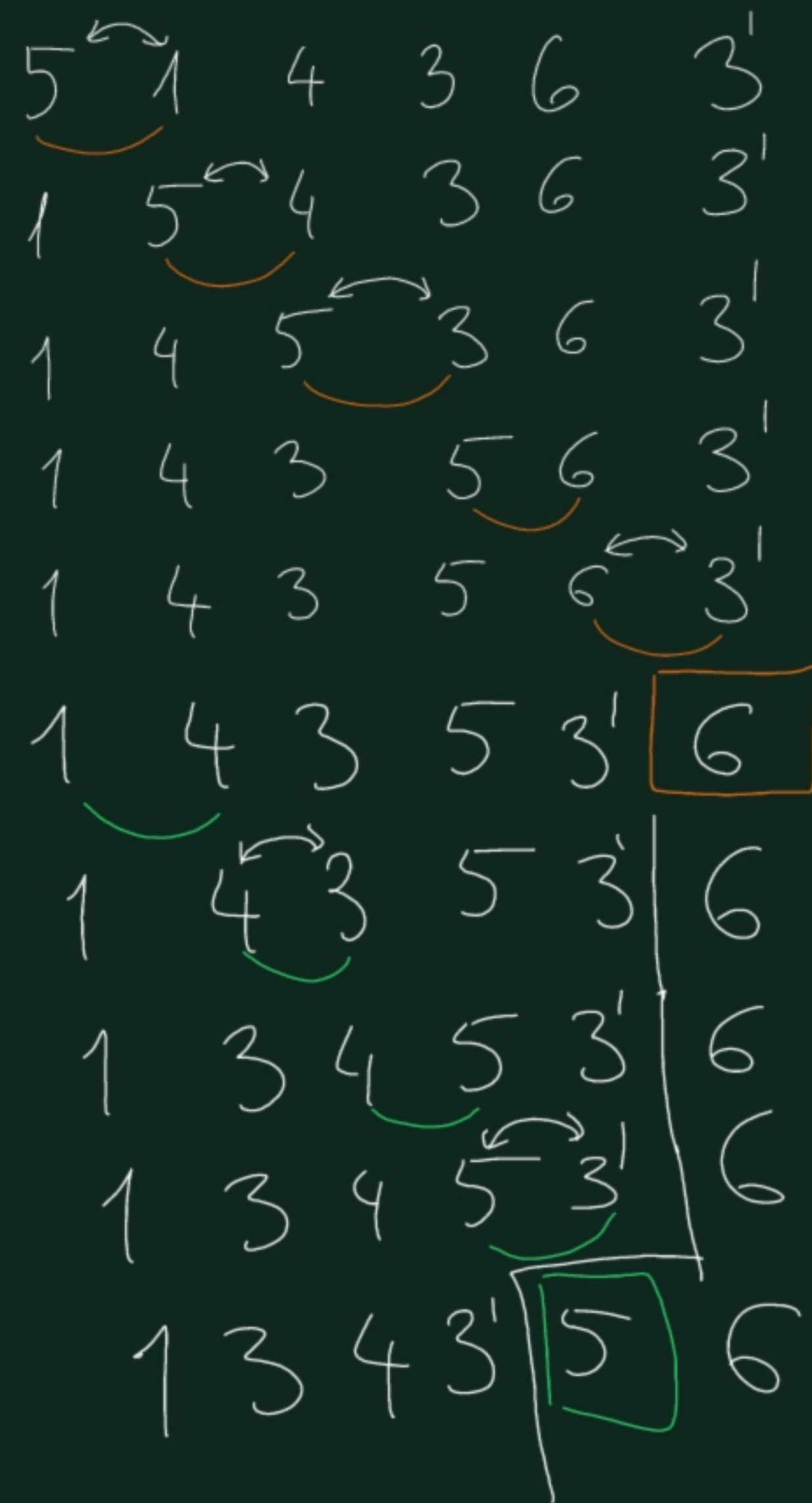
Összefésülő v. Merge Sort

Gyorsrendezés Quick Sort

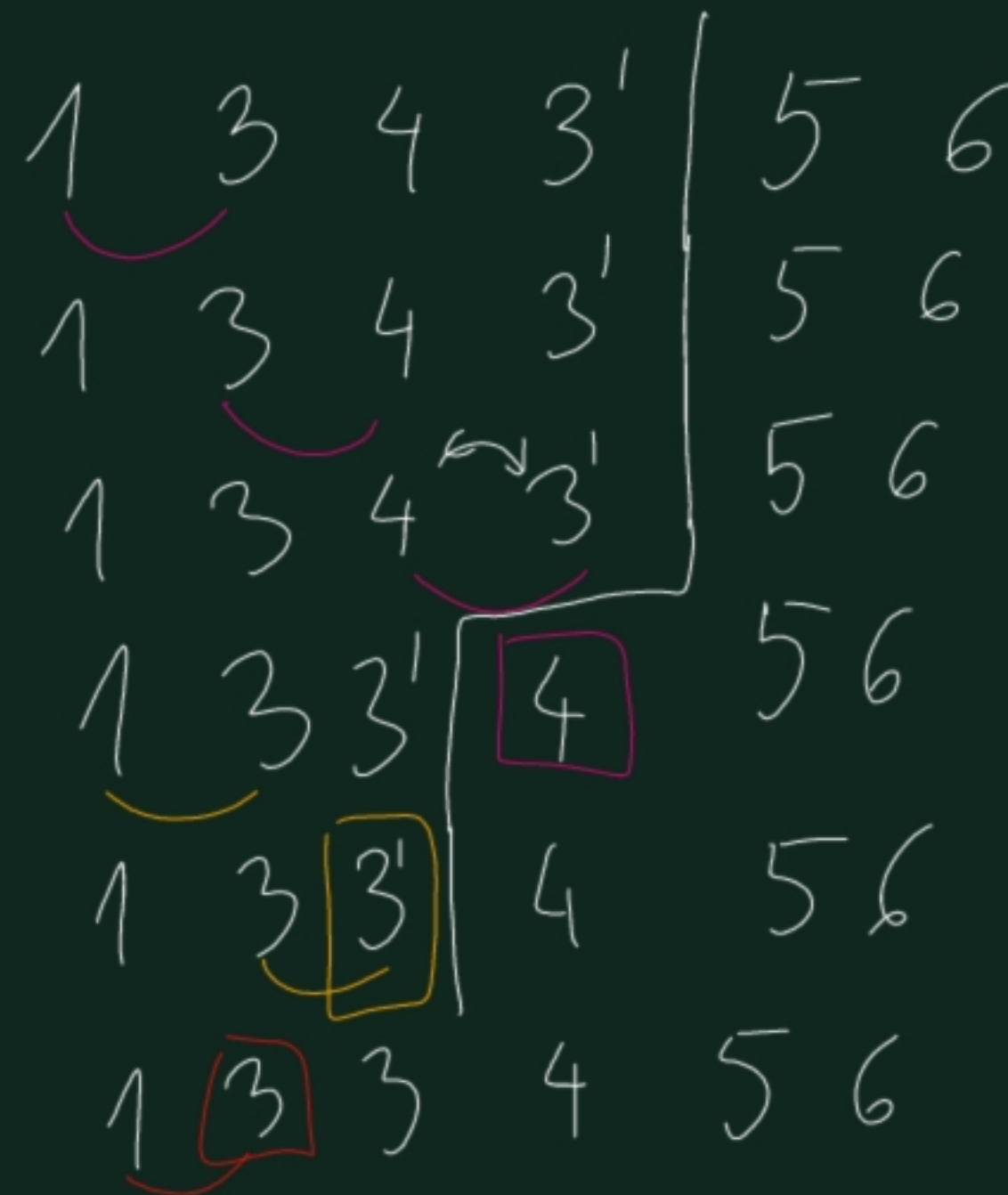
Kupacrendezés Heap Sort



# Buborik renderis



← helyen  
van



BubbleSort( $A : \mathcal{T}[n]$ )

$i = n - 1$  downto 1

$j = 0$  to  $i - 1$

$A[j] > A[j + 1]$

swap ( $A[j], A[j + 1]$ )

SKIP

javítási lehetőségek:

① ha a belső ciklusban nem volt egy cseré sem, akkor a

külső ciklus álljon le! (logikai változó)

→ szorgalmi HF. megírni ezt ↗

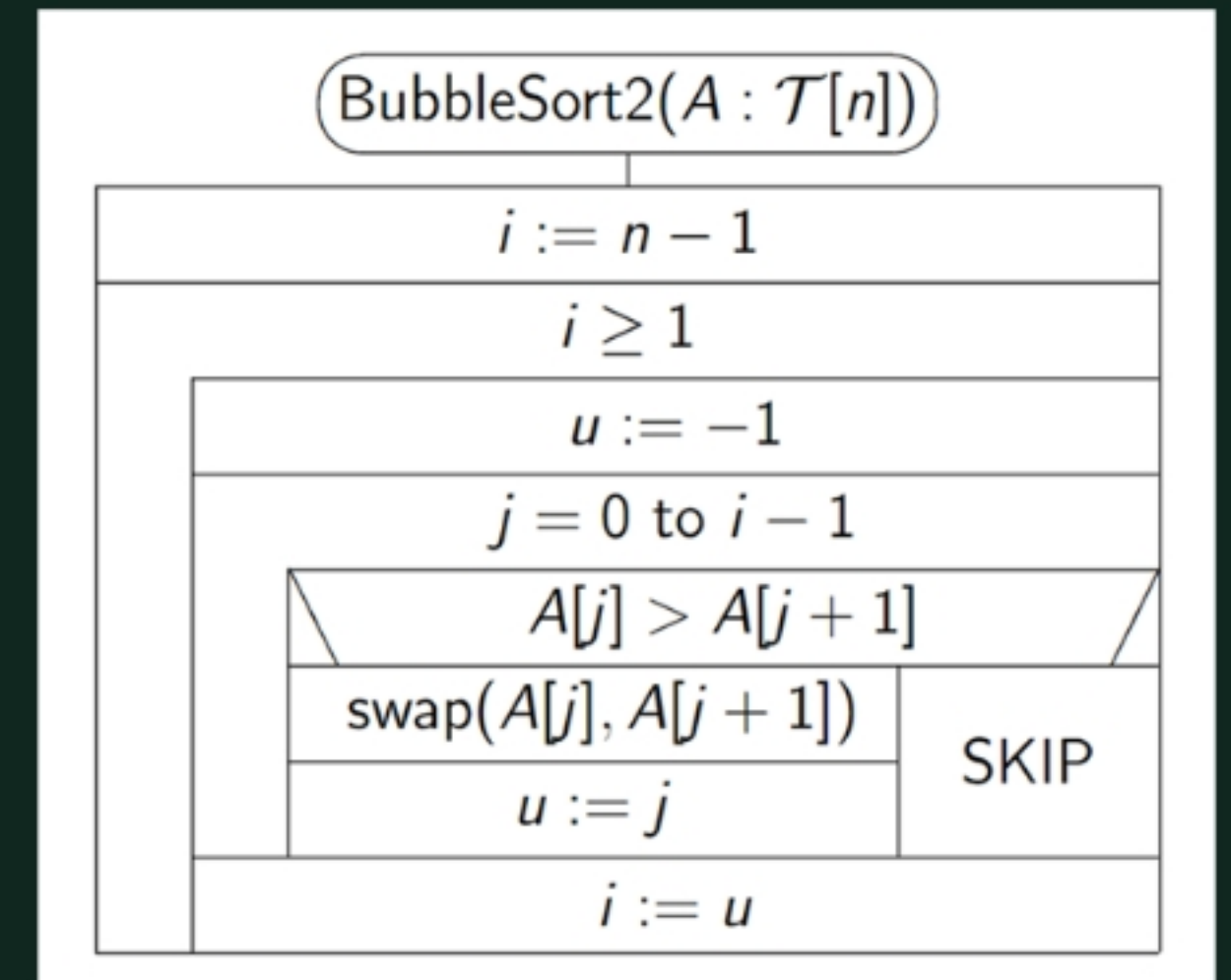
② példa: 

5	1	4	3	6	3	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Az első "menet" végén nem csak a 10-es kerül a helyére, hanem 6-tól 10-ig minden elem.

Honnan tudjuk?  $j=4$  után nem volt cseré...

→ hol volt az utolsó cseré?





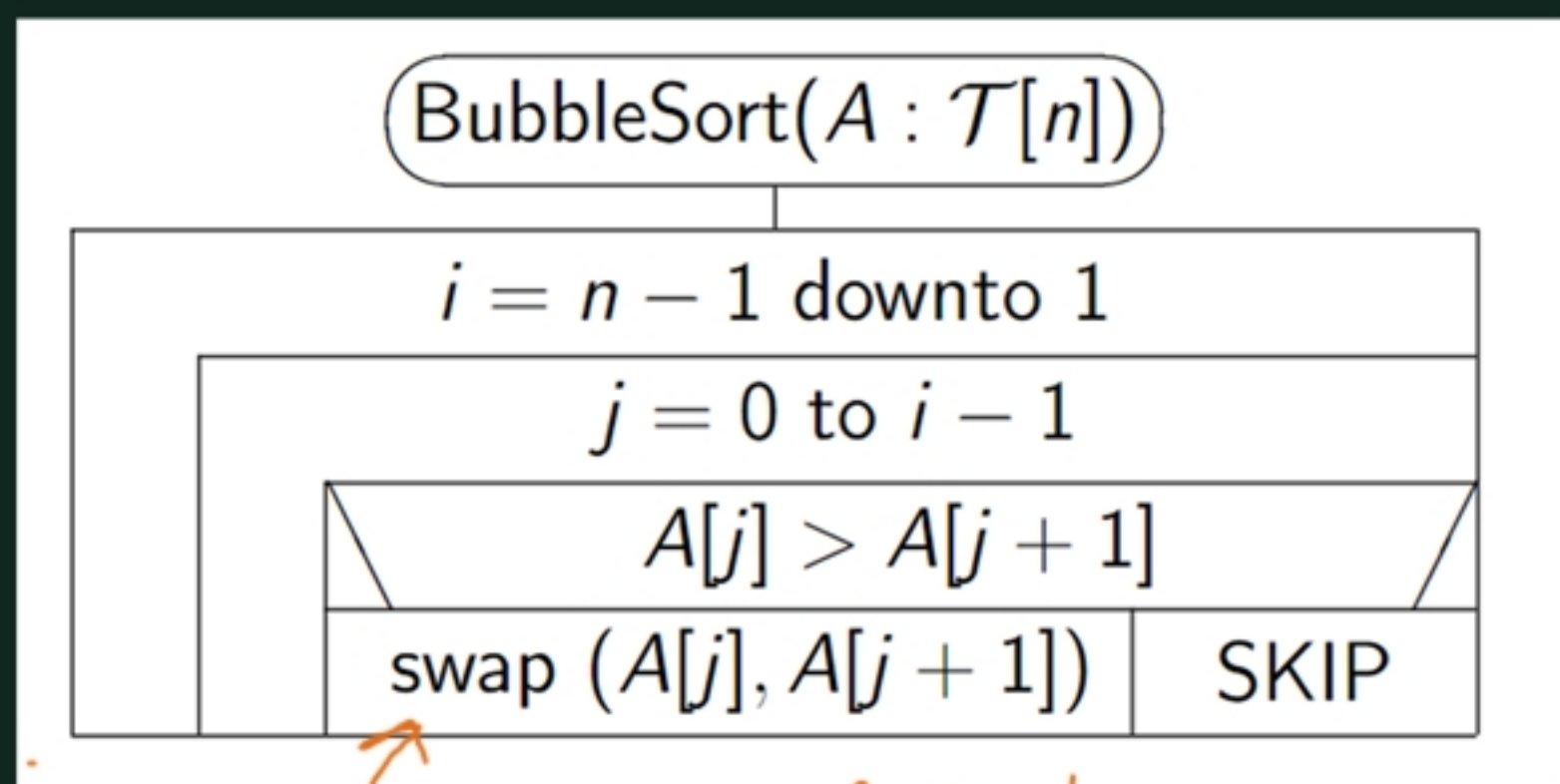
2 3 4  $\hookrightarrow$  1 5

→ ötlet: mezzink, hogy hol volt az  
első csere → kör. ciklusban

1-gyel kombinálva indulunk csak

2 3 4 .... 100 5 3 4 2

meg. HF ert kidolgozni



$n$   
 $n + n - 1 + \dots + 1$   
 $n - 1 + n - 2 + \dots + 1$

ez is eljövís

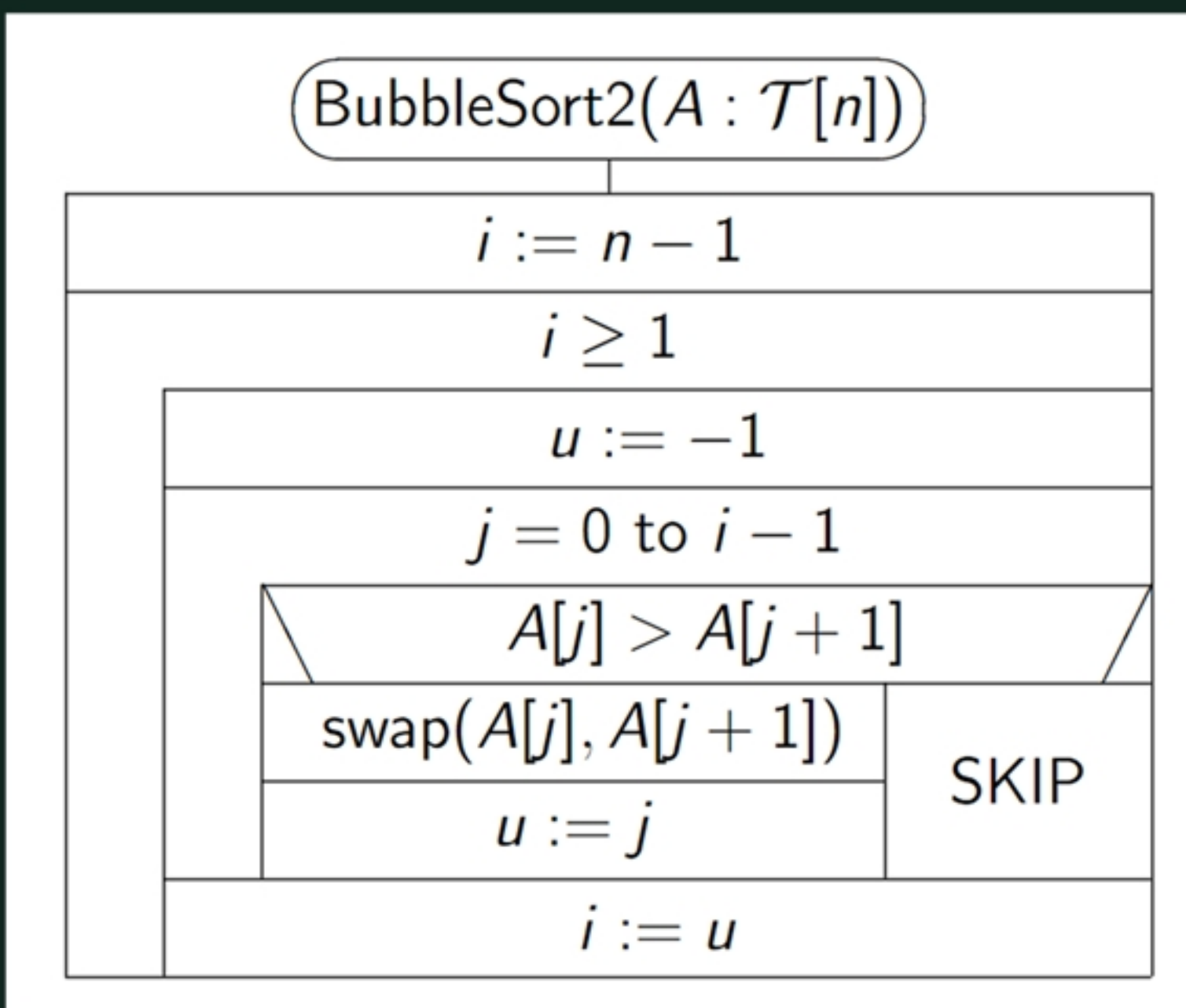
$$m T(n) = 1 + n - 1 + \frac{n(n-1)}{2}$$

$$M T(n) = 1 + n - 1 + \frac{n(n-1)}{2} + \frac{n(n-1)}{2}$$

A T(n)

ciklusit  
 +  
 elj. hív } műveletigény

ennyi  
 swap



jövő" hét

MaxSelectionSort( $A : \mathcal{T}[n]$ )

$i = n - 1$  downto 1

$Maxind := 0$

$j = 1$  to  $i$

$A[j] > A[Maxind]$

$Maxind := j$

SKIP

swap( $A[Maxind]$ ,  $A[i]$ )

