ADATSZERKEZETEK ÉS ALGORITMUSOK

„Rendező algoritmusok”
Négyzetes rendezők

Buborék rendezés

- Feladat: Rendezzük az $A[1 \ldots n]$ vektort!
  A vektor elemtípusa tetszőleges T típus, amire egy teljes rendezés értelmezhető
- Buborék rendezés alapötlete:
  - a vektor elejétől kezdve „felbuborékoltatjuk” a legnagyobb elemet. Utána ugyanezt tesszük az eggyel rövidebb vektorra, stb. Végül, utoljára még az első két elemre is végrehajtjuk a „buborékoltatást”

Buborék rendezés

- Egy sorozat rendezett ⇔ nincs az elemek között inverzió
- Ez a rendezés az inverziók csökkentésével rendez

12	5	7	9	11	10

Buborék rendezés

12	5	7	9	11	10

Buborék rendezés

5	12	7	9	11	10

Buborék rendezés

5	12	7	9	11	10

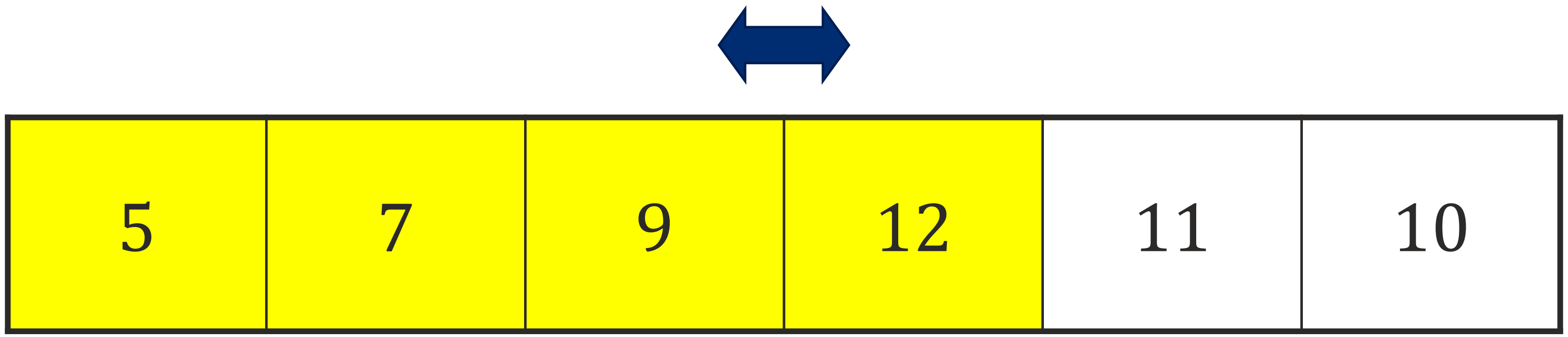
Buborék rendezés

5	7	12	9	11	10

Buborék rendezés

5	7	12	9	11	10

Buborék rendezés



Buborék rendezés

5	7	9	12	11	10

Buborék rendezés

5	7	9	11	12	10

Buborék rendezés

5	7	9	11	12	10

Buborék rendezés

5	7	9	11	10	12

Buborék rendezés

5	7	9	11	10	12

Buborék rendezés

5	7	9	11	10	12

Buborék rendezés

5	7	9	11	10	12

Buborék rendezés

5	7	9	11	10	12

Buborék rendezés

5	7	9	11	10	12

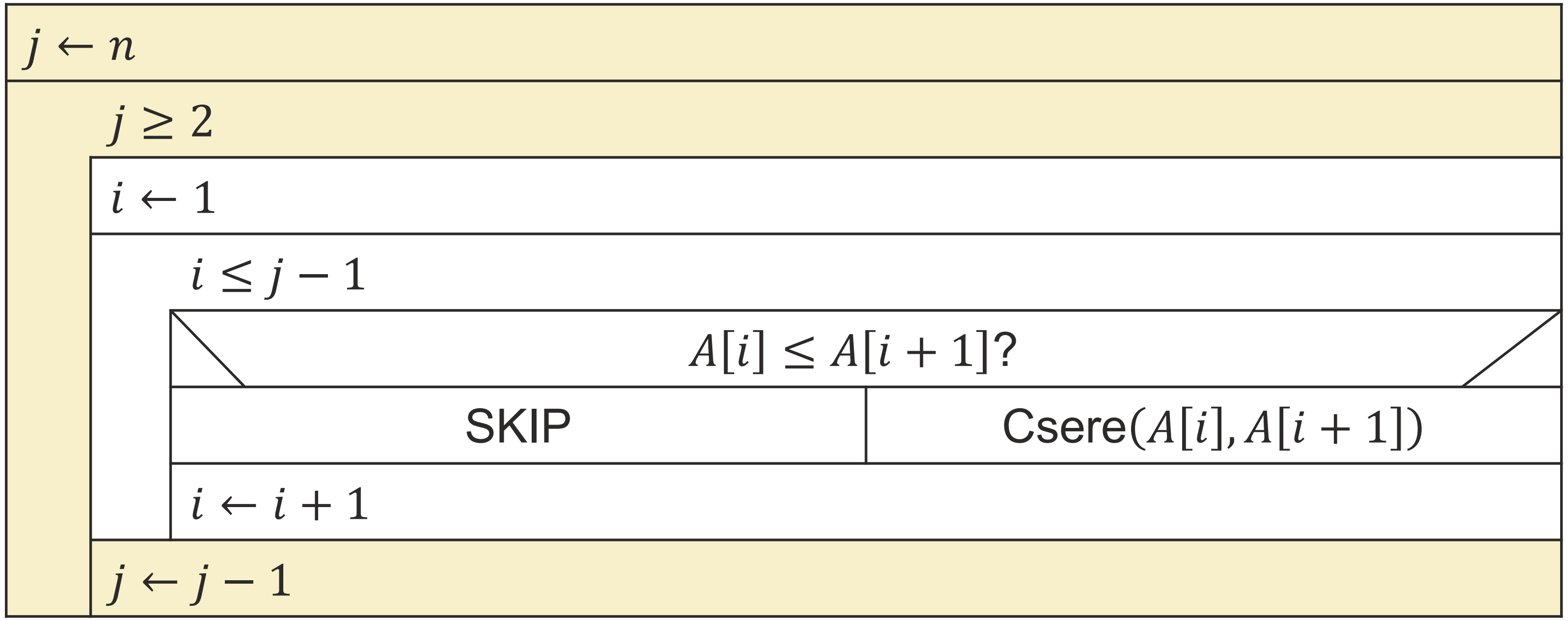
Buborék rendezés

5	7	9	10	11	12

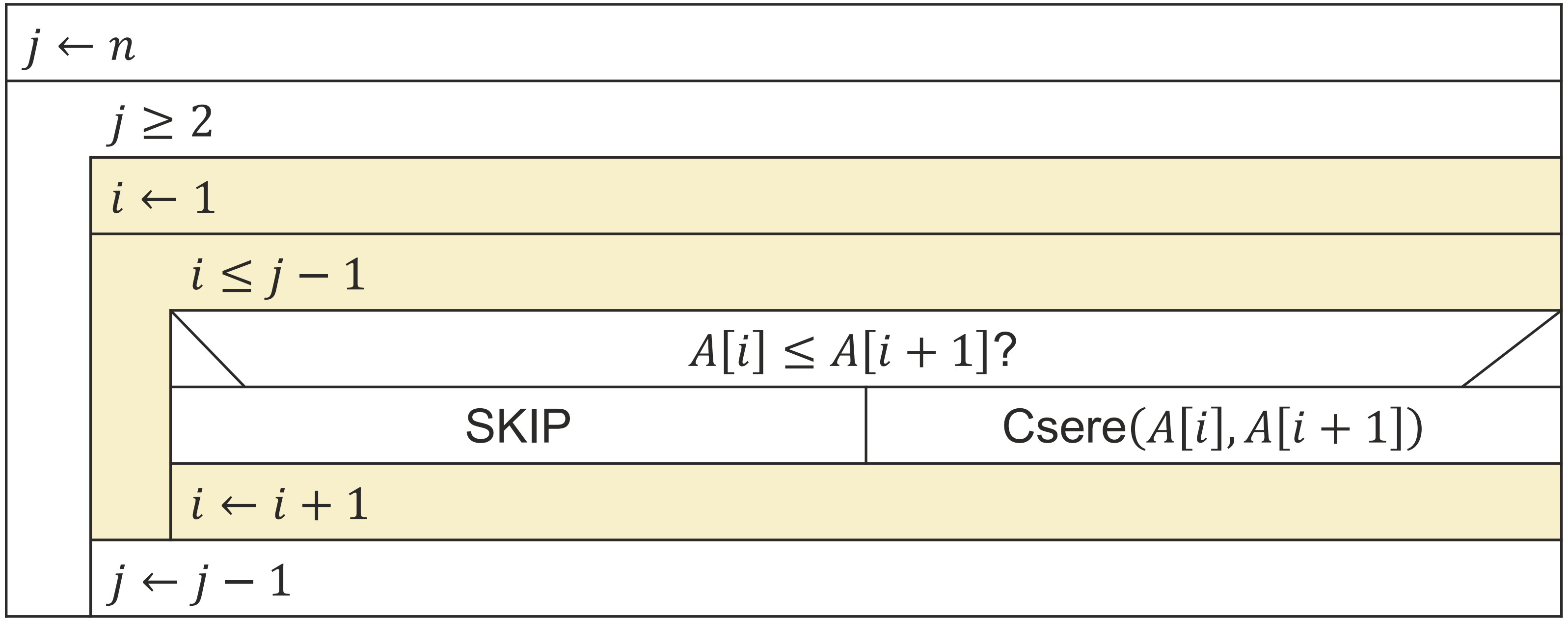
Buborék rendezés

5	7	9	10	11	12

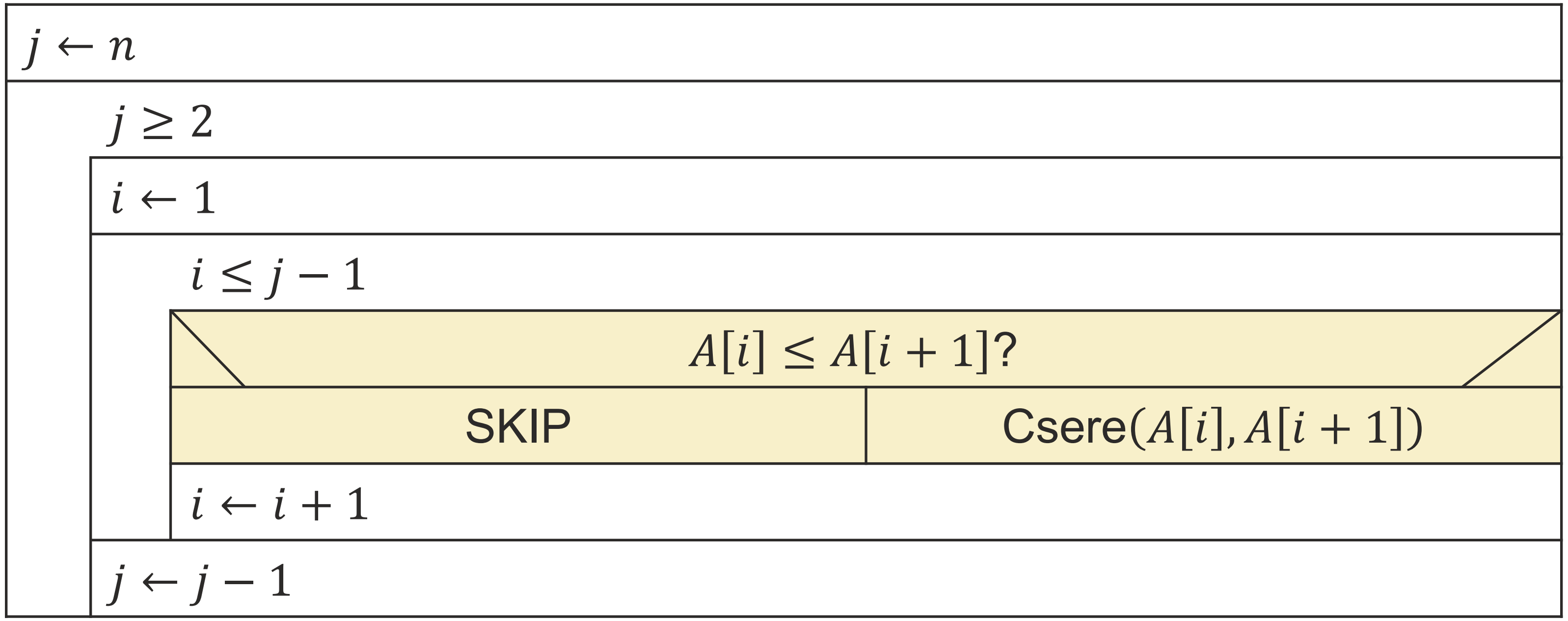
Buborék rendezés algoritmusa



Buborék rendezés algoritmusa

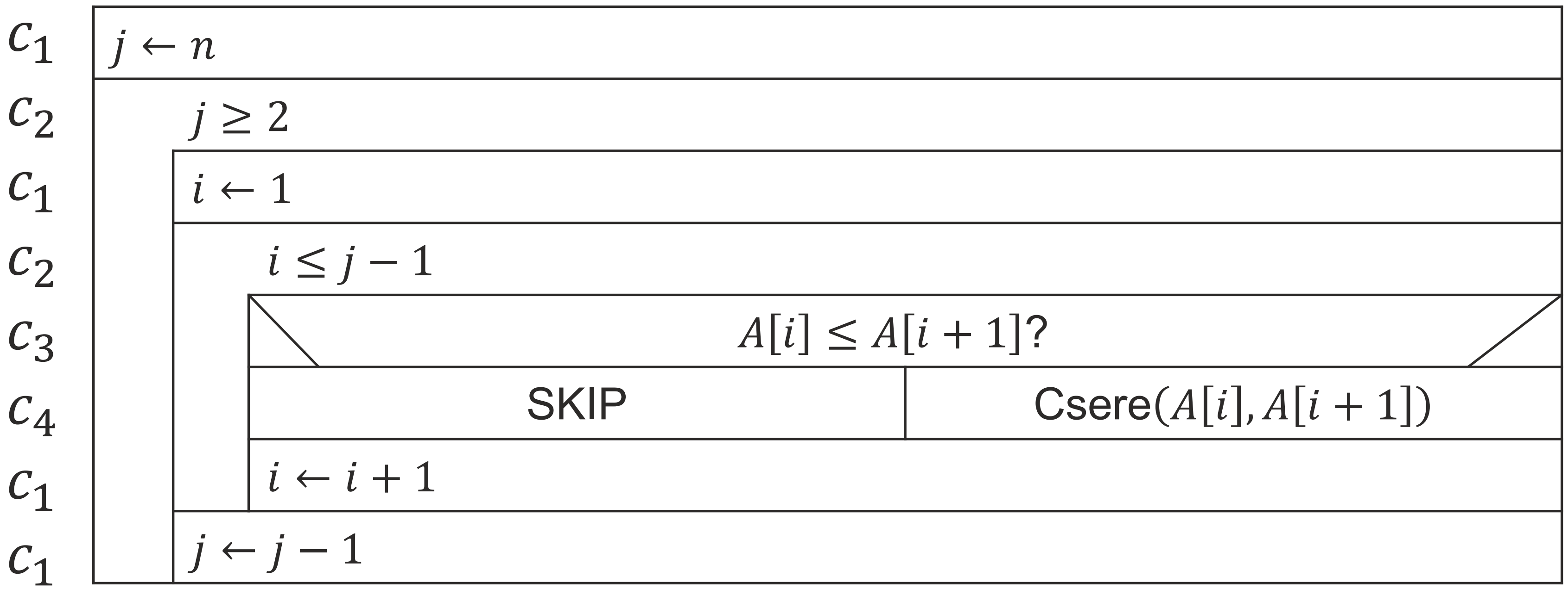


Buborék rendezés algoritmusa



Buborék rendezés algoritmusa

- Műveletek időigénye



Buborék rendezés – időráfordítás

- Külső ciklus
  - $n-1$ –szer fut le, előtte kezdő értékadás
  - A feltételt n-szer ellenőrzi, j-t $n-1$-szer csökkenti
    - $c_1 + n * c_2 + (n-1) * c_1$
- Belső ciklus lefutásainak száma: $n-1, n-2, \dots 2, 1$
  - mindannyiszor 1 kezdőértékadás
  - a ciklusfeltétel eggyel többször ellenőrzi
  - i-t növeljük $(1 + \cdots + \ n-1)$-szer
    - $(n-1) * c_1 + \ (2 + \cdots + n) * c_2 + (1 + \cdots + n - 1) * c_1$
- $A[i]$ és $A[i+1]$ összehasonlításainak száma:
  - $(1 + \cdots + \ n-1) * c_3$
- A cserék száma A-tól függ = A inverzióinak száma
  - Ez 0 és $\binom{n}{2}$ között van
  - $\mathrm{inv}(A) * c_4$

Buborék rendezés – összegezve

- $T(A) = c_1 + n * c_2 + (n-1) * c_1 + (n-1) * c_1 + \ (2 + \cdots + n) * c_2 + (1 + \cdots + n - 1) * c_1 + (1 + \cdots + \ n - 1) * c_3 + \mathrm{inv}(A) * c_4 =$

- $c_1 + n * c_2 + n * c_1 + n * c_1 - c_1 + (n+2) * \frac{n-1}{2} * c_2 + n * \frac{n-1}{2} * c_1 + \ n * \frac{n-1}{2} * c_3 + \mathrm{inv}(A) * c_4 =$

- $n^2 * \left(\frac{c_1}{2} + \frac{c_2}{2} + \frac{c_3}{2}\right) + n * \left(3 * \frac{c_1}{2} + 3 * \frac{c_2}{2} - \frac{c_3}{2}\right) - (c_1 - c_2) + \mathrm{inv}(A) * c_4$

Buborék rendezés

- Néhány egyszerűsítő feltételezés
  1. Feltételezés
     $c_1 \ll c_3$ és $c_2 \ll c_4$
     vagyis az elemek összehasonlítása és cseréje adja a költség javát
     - Ekkor
       $$T(A) \approx n^2 * \frac{c_3}{2} - n * \frac{c_3}{2} + \text{inv}(A) * c_4 = n * \frac{n-1}{2} * c_3 + \text{inv}(A) * c_4$$
  2. Feltételezés:
     $c_3$és $c_4$az egyes gépekre jellemző állandók. Ha történetesen $c_3 \approx c_4$, akkor a végrehajtási időt jellemzi a domináns műveletek száma:
     - $T(A) \approx n * \frac{n-1}{2} + \text{inv}(A)$
       $T(A)$ helyett $T(n)$-t bevezetve, a szokásos írásmód:
       $$T(n) \approx n * \frac{n-1}{2} + \text{inv}(A)$$

Buborék rendezés

- Néhány egyszerűsítő feltételezés
  3. Feltételezés:
     általában külön-külön kérdezzük az egyes műveletek számát:
     - Ö$(n) \approx n * \frac{n-1}{2}$
     - Cs$(n) \approx$ inv(A)
  4. Feltételezés:
     ismerjük a lehetséges bejövő input adatokat, kiszámíthatjuk
     - a legrosszabb $(M\,T(n))$ és
     - a legjobb esetben $(m\,T(n))$
     - átlagos esetben $(A\,T(n))$

Buborék rendezés

- Az előzőekből:
  - Az összehasonlítások száma minden n hosszú vektorra ugyananannyi:
    - $M\,\text{Ö}(n) = m\,\text{Ö}(n) = A\,\text{Ö}(n) = n * \frac{n-1}{2}$
  - A cserék száma:
    - $M\,\text{Cs}(n) \; = \; n * \frac{n-1}{2}$
    - $m\,\text{Cs}(n) \; = \; 0$
    - $A\,\text{Cs}(n) = \; n * \frac{n-1}{4} = M\,\frac{\text{Cs}(n)}{2}$
      - Ez bizonyítható

Buborék rendezés

- Az előzőek pontos értékek, azonban gyakran a költség (műveletszám) nagyságrendje, aszimptotikus viselkedése a hasznos információ
  - $M\,\text{Ö}(n) = \; n * \frac{n-1}{2} = \Theta(n^2)$
  - $M\,\text{Cs}(n) = n * \frac{n-1}{2} = \Theta(n^2)$
  - $A\,\text{Cs}(n) = \; n * \frac{n-1}{4} = M\,\frac{\text{Cs}(n)}{2} = \Theta(n^2)$

Buborék rendezés

- Javított változat
  - Ha nincs már csere, leáll.
    - Cserék száma nem változik
    - $M\ \text{Ö}(n)$ nem változik
    - $m\ \text{Ö}(n)\ =\ n-1$
    - $A\ \text{Ö}(n)\ = ?$ (sejtés: n^2)

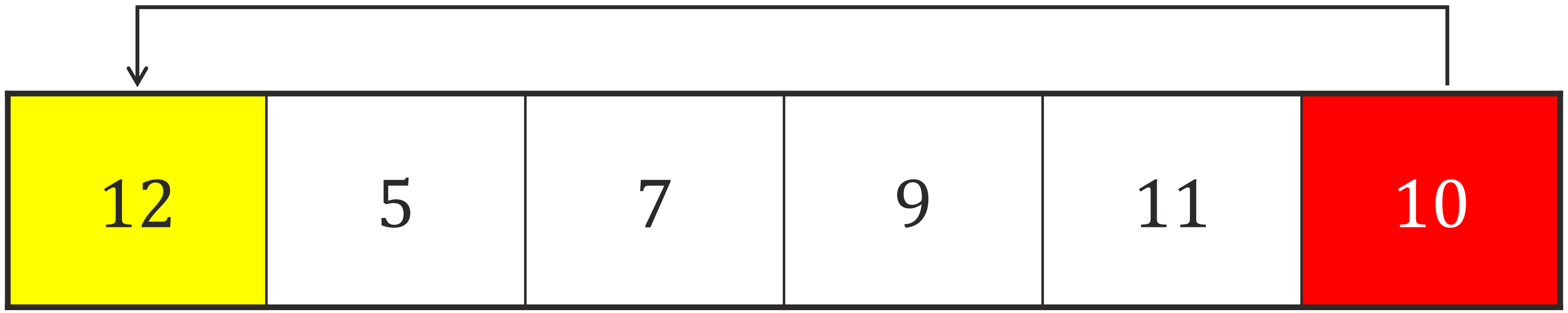
Maximum kiválasztásos rendezés

- Ötlet: feltételezzük, hogy az $A[1..n]$ tömb jobb széle $(j+1..n)$ már rendezve van, minden lépésben kiválasztjuk az $A[1..j]$ résztömb maximális elemét, és kicseréljük a j. helyen lévővel, majd j-t csökkentjük
  - Kezdetben j értéke n.

12	5	7	9	11	10

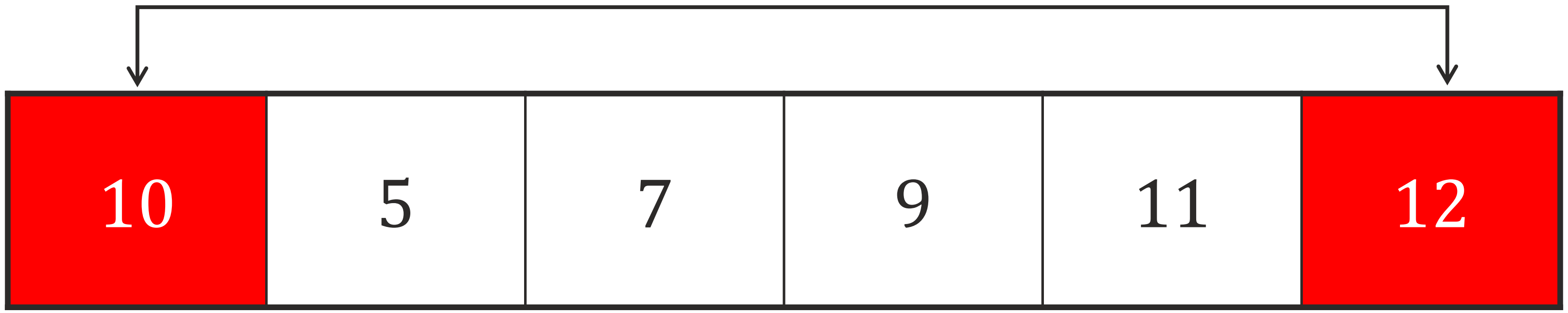
Maximum kiválasztásos rendezés

- Maximum 12
- Hozzá tartozó index 1



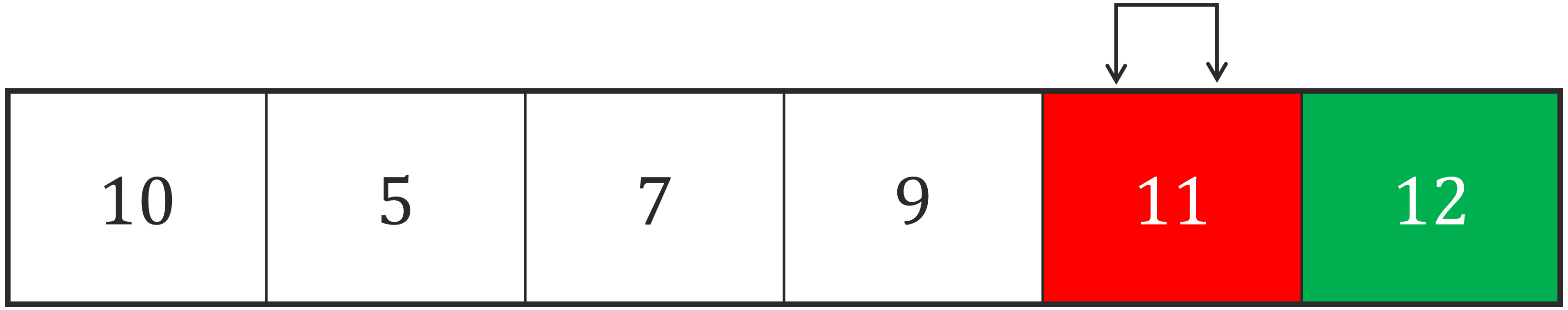
Maximum kiválasztásos rendezés

- Maximum 12
- Hozzá tartozó index 1



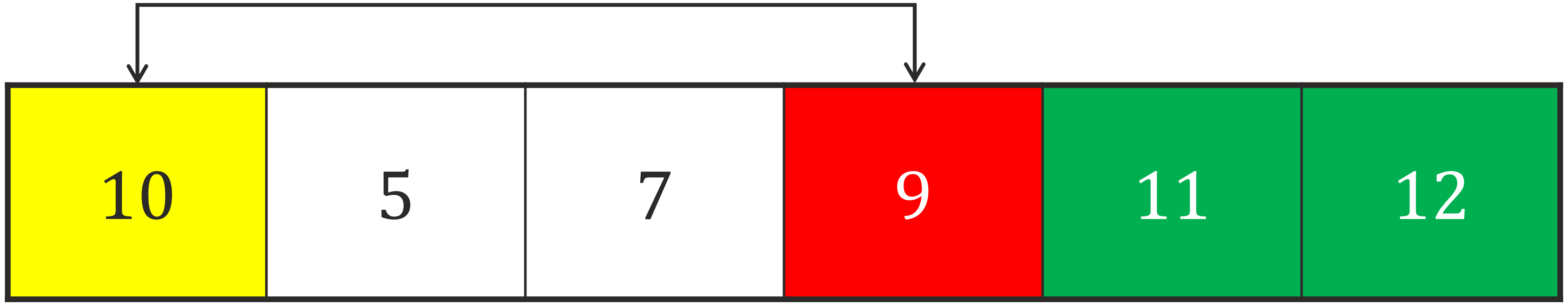
Maximum kiválasztásos rendezés

- Maximum 11
- Hozzá tartozó index 5



Maximum kiválasztásos rendezés

- Maximum 10
- Hozzá tartozó index 1



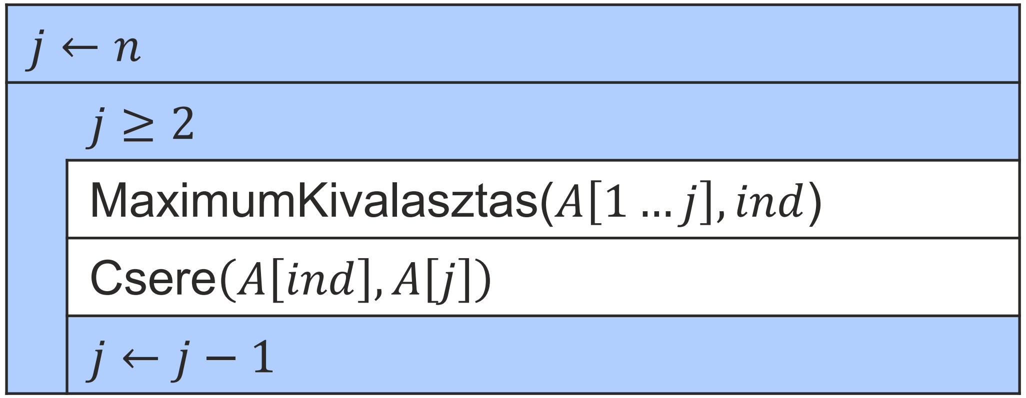
Maximum kiválasztásos rendezés

- Maximum 10
- Hozzá tartozó index 1

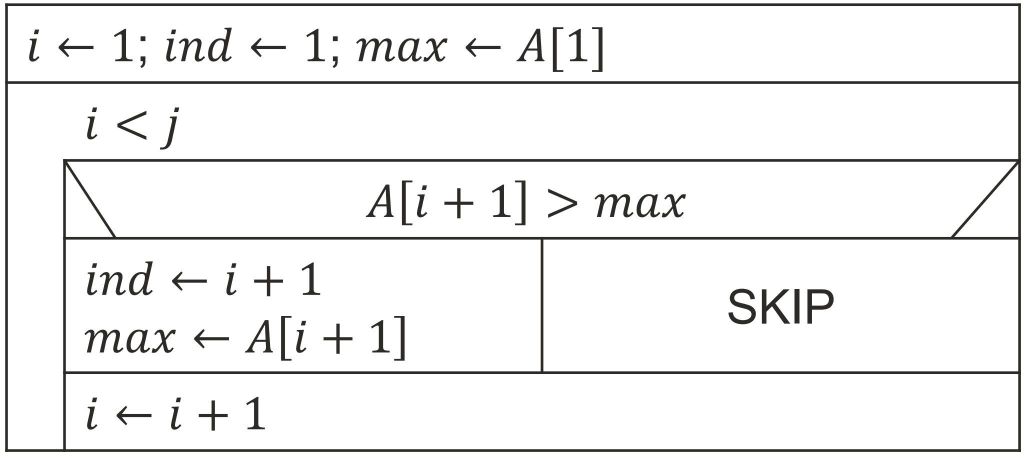
9	5	7	10	11	12

Maximum kiválasztásos rendezés

- A rendező algoritmusa

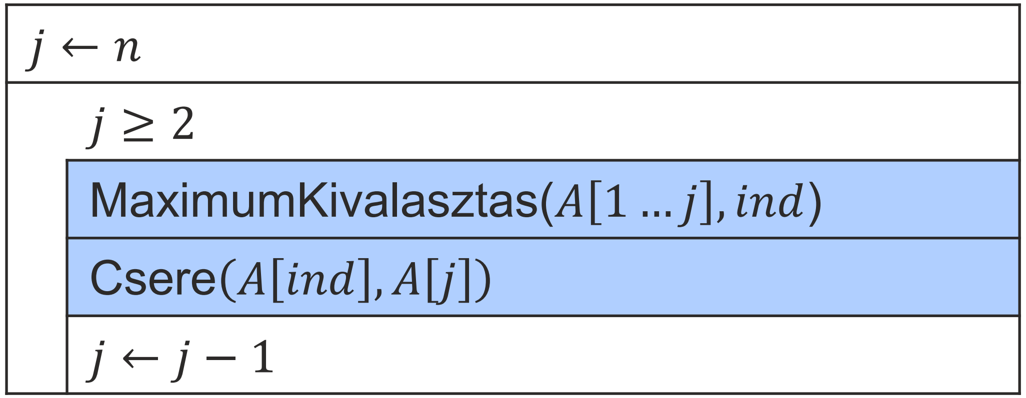


- Maximum kiválasztása

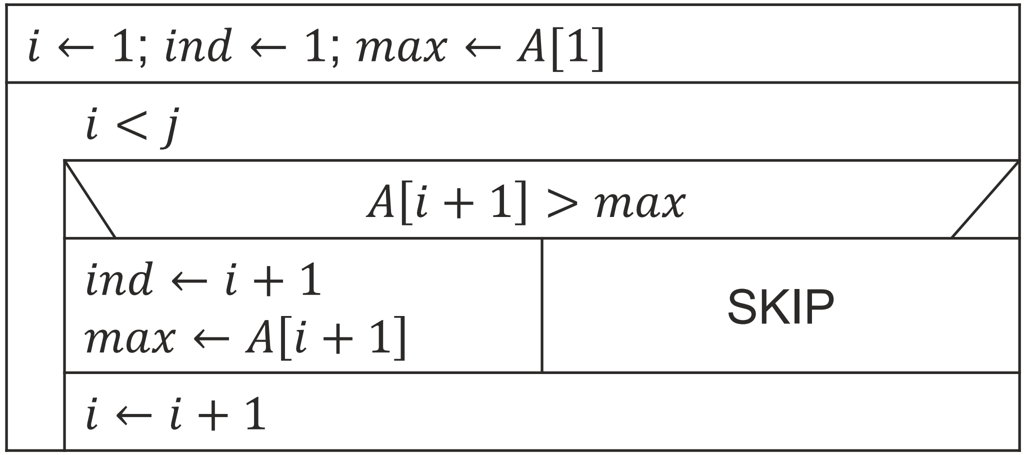


Maximum kiválasztásos rendezés

- A rendező algoritmusa

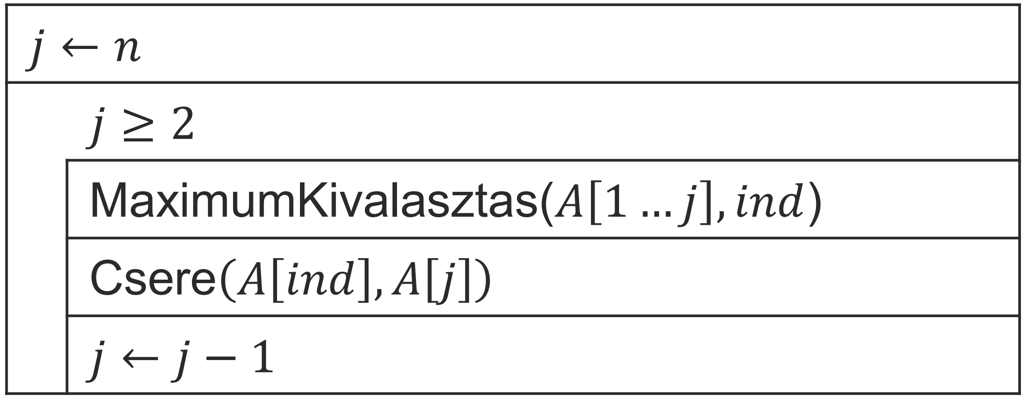


- Maximum kiválasztása

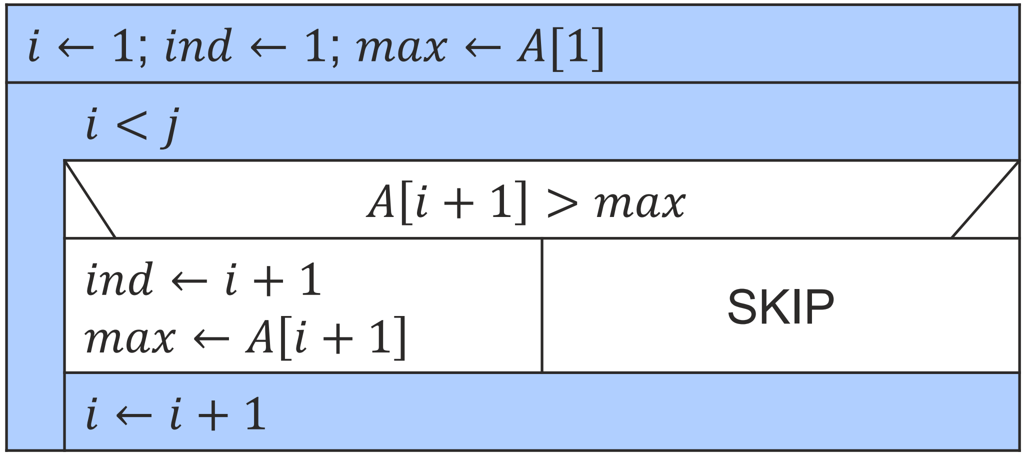


Maximum kiválasztásos rendezés

- A rendező algoritmusa



- Maximum kiválasztása



Maximum kiválasztásos rendezés

- A rendező algoritmusa
  - $M\ddot{O}(n) \geq (n-1) + (n-2) + \ldots + 1 = \Theta(n^2)$

$j \leftarrow n$

$j \geq 2$

- MaximumKivalasztas($A[1 \ldots j], ind$)
- Csere($A[ind], A[j]$)
- $j \leftarrow j - 1$

- Maximum kiválasztása
  - A maximum n elem közül legalább $n-1$ összehasonlítással található meg

$i \leftarrow 1;\ ind \leftarrow 1;\ max \leftarrow A[1]$

$i < j$

- $A[i+1] > max$

$ind \leftarrow i + 1$ $max \leftarrow A[i+1]$	SKIP

- $i \leftarrow i + 1$

Beszúró rendezés

- Egyszerű beillesztéssel egy – egy elemet a helyére viszünk – megkeressük, hogy hová való a már rendezett sorozatban
- Ha tömbben tároljuk, akkor a mozgatások száma is fontos

Beszúró rendezés

- Példa

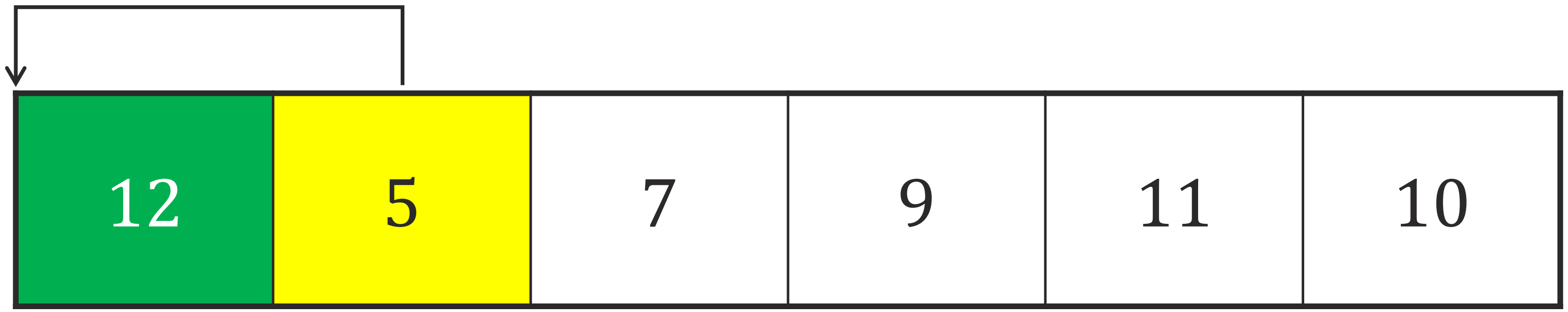
Beszúró rendezés

12	5	7	9	11	10

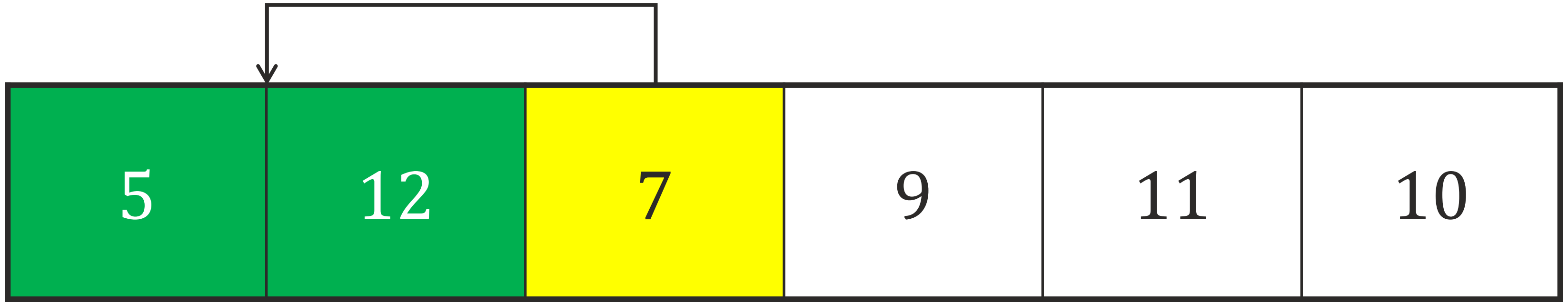
Beszúró rendezés

12	5	7	9	11	10

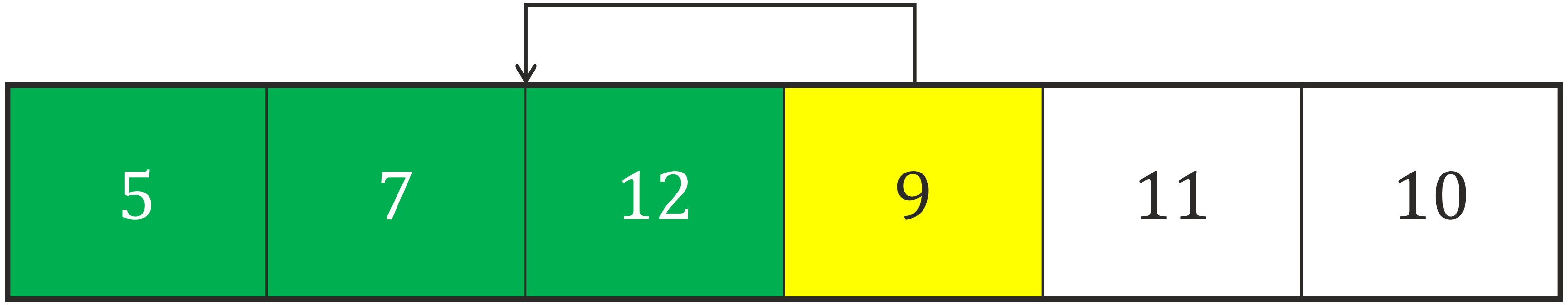
Beszúró rendezés



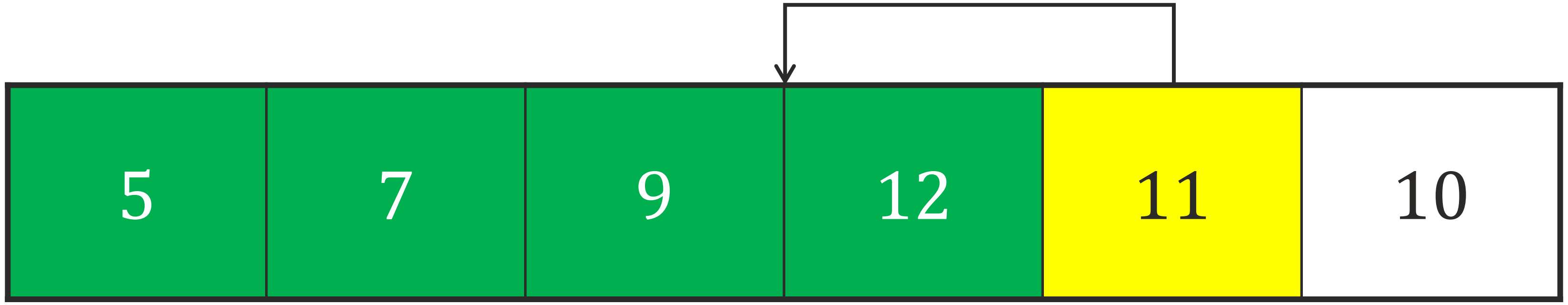
Beszúró rendezés



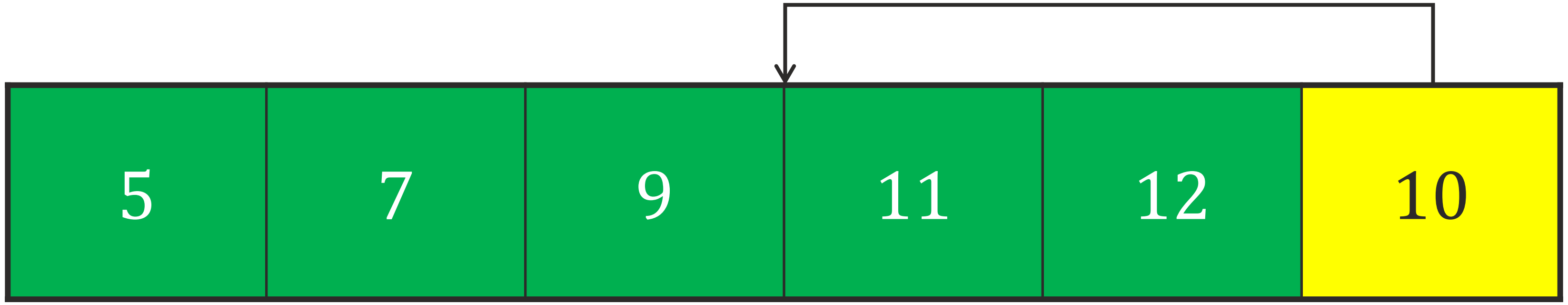
Beszúró rendezés



Beszúró rendezés



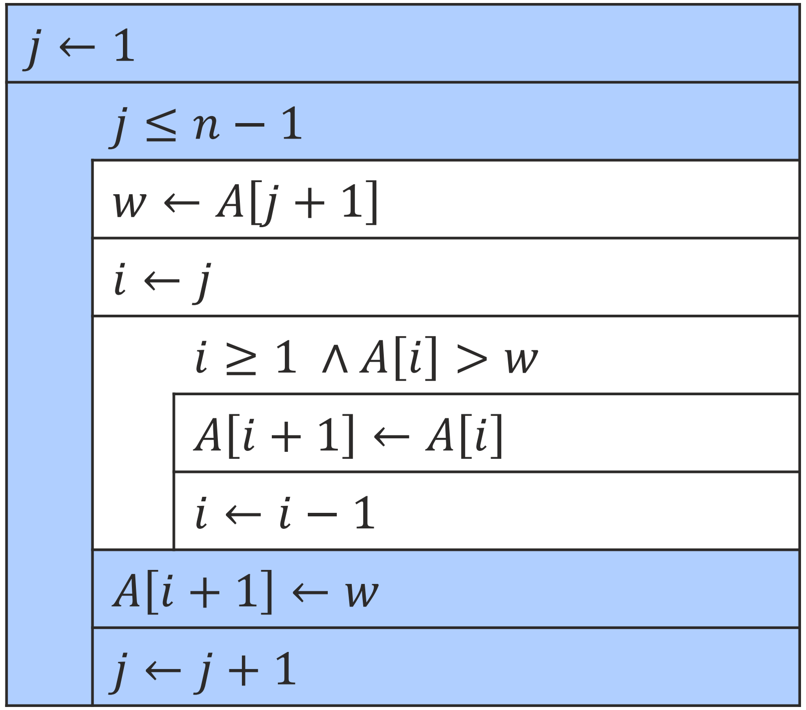
Beszúró rendezés



Beszúró rendezés

5	7	9	10	11	12

Beszúró rendezés algoritmusa



Beszúró rendezés algoritmusa

$j \leftarrow 1$

$j \leq n - 1$

- $w \leftarrow A[j + 1]$
- $i \leftarrow j$
- $i \geq 1 \ \wedge A[i] > w$
  - $A[i + 1] \leftarrow A[i]$
  - $i \leftarrow i - 1$
- $A[i + 1] \leftarrow w$
- $j \leftarrow j + 1$

Beszúró rendezés

- Műveletigény
  - Összehasonlítások
    - $M\ \text{Ö}(n) = n * \frac{n-1}{2} = \Theta(n^2)$
    - $A\ \text{Ö}(n) \approx M\frac{\text{Ö}(n)}{2} = \Theta(n^2)$
    - $m\ \text{Ö}(n) = n - 1$
  - M a mozgatás művelete:
    - $M\ M(n) = (n+2) * \frac{n-1}{2} = \Theta(n^2)$
    - $A\ M(n) = \frac{n^2}{4} = \Theta(n^2)$
    - $m\ M(n) = 2 * (n-1) = \Theta(n)$