

Optimalitatea algoritmilor



Optimalitatea algoritmilor

- ▶ Exemplul 2. Să se determine minimul și maximul elementelor unui vector.

```
if n impar then  $m \leftarrow a_1$ ;  $M \leftarrow a_1$ ;  $k \leftarrow 1$ 
else if  $a_1 < a_2$  then  $m \leftarrow a_1$ ;  $M \leftarrow a_2$ 
    else  $m \leftarrow a_2$ ;  $M \leftarrow a_1$ ;
     $k \leftarrow 2$ 
while  $k \leq \mathbf{n-2}$ 
    if  $a_{k+1} < a_{k+2}$  then if  $a_{k+1} < m$  then  $m \leftarrow a_{k+1}$ 
        if  $a_{k+2} > M$  then  $M \leftarrow a_{k+2}$ 
    else if  $a_{k+2} < m$  then  $m \leftarrow a_{k+2}$ 
        if  $a_{k+1} > M$  then  $M \leftarrow a_{k+1}$ 
     $k \leftarrow k+2$ 
```

- ▶ $T(n)=?$

Optimalitatea algoritmilor

▶ $T(n) = \lceil 3n/2 \rceil - 2$:

◦ n impar :

$$T(n) =$$

◦ n par:

$$T(n) =$$

Optimalitatea algoritmilor

► $T(n) = \lceil 3n/2 \rceil - 2$:

◦ n impar :

$$T(n) = 3(n-1)/2 = (3n+1)/2 - 2 = \lceil 3n/2 \rceil - 2$$

◦ n par:

$$T(n) = 1 + 3(n-2)/2 = 3n/2 - 2 = \lceil 3n/2 \rceil - 2$$

Optimalitatea algoritmilor

- ▶ $T(n) = \lceil 3n/2 \rceil - 2$

- Optimal?

Optimalitatea algoritmilor

- ▶ $T(n) = \lceil 3n/2 \rceil - 2$
- ▶ Proprietate Orice algoritm de determinare a minimului și maximului unui vector cu n elemente bazat pe comparații necesită cel puțin $\lceil 3n/2 \rceil - 2$ comparații
- ▶ Idee de demonstrație:

Optimalitatea algoritmilor

- ▶ $T(n) = \lceil 3n/2 \rceil - 2$
- ▶ Proprietate Orice algoritm de determinare a minimului și maximului unui vector cu n elemente bazat pe comparații necesită cel puțin $\lceil 3n/2 \rceil - 2$ comparații
- ▶ Idee de demonstrație:
 - La un pas al algoritmului analizăm:
 - ce tipuri de elemente pot apărea în funcție de rezultatele comparațiilor deja efectuate
 - între ce tipuri de elemente va face comparații un algoritm eficient

Optimalitatea algoritmilor

Detalii

- ▶ A = mulțimea elementelor care **nu au participat încă la comparări**; $a = |A|$;
- ▶ B = au participat la comparări și **au fost totdeauna mai mari** decât elementele cu care au fost comparate;
 $b = |B|$;
- ▶ C = au participat la comparări și au fost **totdeauna mai mici** decât elementele cu care au fost comparate;
 $c = |C|$;
- ▶ D = au participat la comparări și au fost **cel puțin o dată mai mari și cel puțin o dată mai mici** decât elementele cu care au fost comparate;
 $d = |D|$;

Optimalitatea algoritmilor

Detalii

- ▶ A = mulțimea elementelor care **nu au participat încă** la comparați; $a = |A|$;
- ▶ B = au participat la comparați și **au fost totdeauna mai mari** decât elementele cu care au fost comparate; $b = |B|$;
- ▶ C = au participat la comparați și au fost **totdeauna mai mici** decât elementele cu care au fost comparate; $c = |C|$;
- ▶ D = au participat la comparați și au fost **cel puțin o dată mai mari și cel puțin o dată mai mici** decât elementele cu care au fost comparate; $d = |D|$;



Ce tipuri de elemente nu mai trebuie luate în considerare în comparați într-un algoritm eficient?

Optimalitatea algoritmilor

Detalii

- ▶ A = mulțimea elementelor care **nu au participat** încă la comparări; $a = |A|$;
- ▶ B = au participat la comparări și **au fost totdeauna mai mari** decât elementele cu care au fost comparate; $b = |B|$;
- ▶ C = au participat la comparări și **au fost totdeauna mai mici** decât elementele cu care au fost comparate; $c = |C|$;
- ▶ D = au participat la comparări și au fost **cel puțin o dată mai mari și cel puțin o dată mai mici** decât elementele cu care au fost comparate; $d = |D|$; \Rightarrow **nu pot fi min sau max**

Optimalitatea algoritmilor

Detalii

- ▶ **Un pas al algoritmului** \equiv configurația (a, b, c, d)
- ▶ **Inițial configurația** $(n, 0, 0, 0)$
- ▶ **Final configurația** $(0, 1, 1, n-2)$
- ▶ **Numărul minim de comparații efectuate de algoritm** =
numărul minim de comparații pentru a trece din
 $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$



În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Optimalitatea algoritmilor

- ▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$
- ▶ În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Tipul elementelor comparate	Configurație obținută
A, A	
B, B	
C, C	
A, B	
A, C	
B, C	

Optimalitatea algoritmilor

- ▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$
- ▶ În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	
C, C	
A, B	
A, C	
B, C	

Optimalitatea algoritmilor

- ▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$
- ▶ În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	
A, C	
B, C	

Optimalitatea algoritmilor

- ▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$
- ▶ În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	$(a-1, b, c+1, d)$ – defavorabil $(a-1, b, c, d+1)$
A, C	
B, C	

Optimalitatea algoritmilor

- ▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$
- ▶ În ce configurații se poate ajunge din (a, b, c, d) în urma unei comparații?

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	$(a-1, b, c+1, d)$ – defavorabil $(a-1, b, c, d+1)$
A, C	$(a-1, b+1, c, d)$ – defavorabil $(a-1, b, c, d+1)$
B, C	(a, b, c, d) – defavorabil $(a, b-1, c-1, d+2)$

Optimalitatea algoritmilor

► $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	$(a-1, b, c+1, d)$ – defavorabil
A, C	$(a-1, b+1, c, d)$ – defavorabil
B, C	(a, b, c, d) – defavorabil

► Pentru $n=2k$, succesiunea minimă de la $(n, 0, 0, 0)$ la $(0, 1, 1, n-2)$:

Optimalitatea algoritmilor

▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	$(a-1, b, c+1, d)$ – defavorabil
A, C	$(a-1, b+1, c, d)$ – defavorabil
B, C	(a, b, c, d) – defavorabil

▶ **Pentru $n=2k$, succesiunea minimă de la $(n, 0, 0, 0)$ la $(0, 1, 1, n-2)$:**

- k comparații între perechi de elemente din A $\Rightarrow (0, k, k, 0)$
- prin $k-1$ comparații B, B $\Rightarrow (0, 1, k, k-1)$
- prin $k-1$ comparații C, C $\Rightarrow (0, 1, 1, n-2)$

Total $k + (k-1) + (k-1) = 3k-2 = \lceil 3n/2 \rceil - 2$ comparații.

Optimalitatea algoritmilor

▶ $(n, 0, 0, 0) \rightarrow (0, 1, 1, n-2)$

Tipul elementelor comparate	Configurație obținută
A, A	$(a-2, b+1, c+1, d)$
B, B	$(a, b-1, c, d+1)$
C, C	$(a, b, c-1, d+1)$
A, B	$(a-1, b, c+1, d)$ – defavorabil
A, C	$(a-1, b+1, c, d)$ – defavorabil
B, C	(a, b, c, d) – defavorabil

▶ **Pentru $n=2k+1$ – exercițiu**