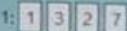


Test sortări

Fie un tablou continand cheile:

3; 1; 7; 2

Sa se scrie configuratia taboului dupa fiecare pas al sortarii prin amestecare (ShakerSort), considerand ca parcurgerea se face de la dreapta la stanga in primul for si de la stanga la dreapta in urmatorul for si un pas contine o parcurgere de la stanga la dreapta sau de la dreapta la stanga.

pas 1: 

pas 2: 

	0	1	2	3
initial	3	1	4	2
	$i = 4 - 2 = 2$	[2]	[3]	

I. comparăm 7 și 2
 >

⇒ schimbăm

3	1	2	7
$i = 1$	[1]	[2]	

comparăm 1 și 2
 <

3	1	2	7
$i = 0$	[0]	[1]	
comparăm	3	1	
	>		
1	3	2	7

am ieșit din primul for

II. intrăm în al doilea for

$i = 0$	[0]	[1]
comparăm	1	> 3

1	3	2	7
$i = 1$	[1]	[2]	

comparăm 3 > 2

1	2	3	7
$i = 2$	[2]	[3]	

comparăm 3 > 7

1	2	3	7

```
#include <stdio.h>

void shakerSort(int arr[], int n) {
    int swapped = 1;
    while (swapped) {
        swapped = 0;
        // Parcurgere de la dreapta la stanga
        for (int i = n - 2; i >= 0; i--) {
            if (arr[i] > arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
                swapped = 1;
            }
        }
        // Dacă nu s-au făcut schimbări, lista este sortată
        if (!swapped) break;
        // Resetez flag-ul swapped pentru parcurgerea de la stanga la dreapta
        swapped = 0;
        // Parcurgere de la stanga la dreapta
        for (int i = 0; i < n - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
                swapped = 1;
            }
        }
    }
    int main() {
        int arr[] = {3, 1, 7, 2};
        int n = sizeof(arr) / sizeof(arr[0]);
        shakerSort(arr, n);

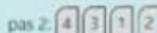
        printf("Tabloul sortat este:\n");
        for (int i = 0; i < n; i++) {
            printf("%d ", arr[i]);
        }
        return 0;
    }
}
```

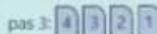
Fie un tablou continand cheile:

1; 4; 3; 2

Sa se scrie configuratia tabloului dupa fiecare pas al sortarii in mod **descrescator** folosind sortarea prin interschimbare (BubbleSort), considerand ca parcurgerea se face de la stanga la dreapta si un pas contine doar o comparatie si maxim o interschimbare.

pas 1: 

pas 2: 

pas 3: 

$v[i] < v[i+1]$

→ swap

0 1 2 3

initial

1 4 3 2

I. $i=0 \rightarrow j=0$ $v[0]$ si $v[1]$

comparăm 1 si 4

facem swap

4 1 3 2

II.

$j=1$ $v[1]$ si $v[2]$

comparăm 1 si 3

facem swap

4 3 1 2

III.

$j=2$ $v[2]$ si $v[3]$

comparăm 1 si 2

facem swap

4 3 2 1

```
void bubble_sort(tip_element a[], int n)
```

```
{
```

```
int i, j;
```

```
for (i = 0; i < n - 1; i++)
```

```
{
```

```
for (j = n - 1; j > i; j--)
```

```
if (a[j].cheie < a[j - 1].cheie) //daca nu sunt in ordinea potrivita
```

```
swap(&a[j], &a[j - 1]);
```

```
}
```

```
}
```

```
void bubbleSort(int arr[], int n){
```

```
int i, j;
```

```
bool swapped;
```

```
for (i = 0; i < n - 1; i++) {
```

```
swapped = false;
```

```
for (j = 0; j < n - i - 1; j++) {
```

```
if (arr[j] > arr[j + 1]) {
```

```
swap(&arr[j], &arr[j + 1]);
```

```
}
```

```
swapped = true;
```

```
}
```

```
}
```

```
// If no two elements were swapped by inner loop,
```

```
// then break
```

```
if (swapped == false)
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

Fie un tablou continand cheile:

3; 1; 7; 2

Sa se scrie configuratia tabloului dupa fiecare pas al sortarii in mod **descrescator** folosind sortare (BubbleSort). Se considera ca parcurgerea se face de la stanga la dreapta si un pas contine o comparatie si maxim o intereschimbare.

pas 1:

pas 2:

pas 3:

Fie un tablou continand cheile:

3; 1; 7; 2

Sa se scrie configuratia tabloului dupa fiecare pas al sortarii prin interschimbare (BubbleSort), considerand ca parcurgerea se face de la stanga la dreapta si un pas contine o comparatie si maxim o interschimbare.

pas 1:

1	3	7	2
---	---	---	---

pas 2:

1	3	7	2
---	---	---	---

pas 3:

1	3	7	2
---	---	---	---

Fie un tablou continand cheile:

8; 1; 7; 2

Sa se scrie configuratia tabloului dupa fiecare pas al sortarii in mod **descrescator** folosind sortarea prin interschimbare (BubbleSort). Se considera ca parcurgerea se face de la stanga la dreapta si un pas contine o comparatie si maxim o interschimbare.

pas 1:

1	3	7	2
---	---	---	---

pas 2:

1	3	7	2
---	---	---	---

pas 3:

1	3	7	2
---	---	---	---

Fie un tablou continand cheile:

2; 3; 7; 1

Sa se scrie configuratia tabloului dupa primii pasi ai sortarii prin selectie (SelectSort), considerand ca un pas este definit de adaugarea unui element la sirul ordonat:

pas 1:

pas 2:

pas 3:

void selection_sort(tip_element a[], int n)

```
{  
    int i, j, min; // min retine INDEXUL elementului cu valoare minima  
    for (i = 0; i < n - 1; i++)  
    {  
        min = i;  
        for (j = i + 1; j < n; j++)  
        {  
            if (a[j].cheie < a[min].cheie)  
                min = j;  
        }  
        swap(&a[min], &a[i]);  
    }  
}
```

	0	1	2	3
initial	2	3	7	1

I i=0 min = 0

căutăm min → min = 3

dăm swap v[0] cu v[3]

1 3 7 2

II. i=1 → min = 3

dăm swap

1 2 7 3

III. i=2 → min = 3

dăm swap

1 2 3 7

Fie un tablou continand cheile:

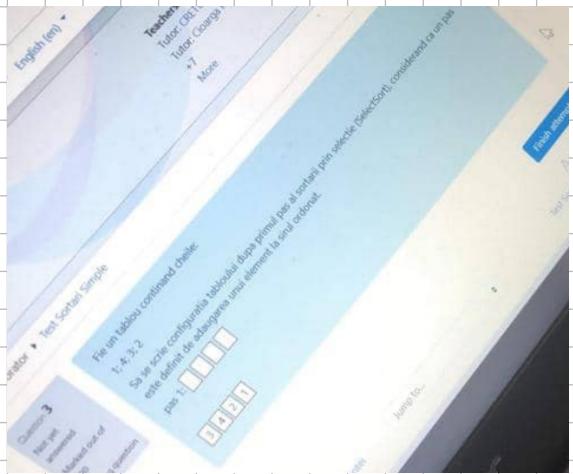
5; 7; 1; 3

Sa se scrie configuratia tabloului dupa primii pasi ai sortarii prin selectie (SelectSort), considerand ca un pas este definit de adaugarea unui element la sirul ordonat.

pas 1: 

pas 2: 

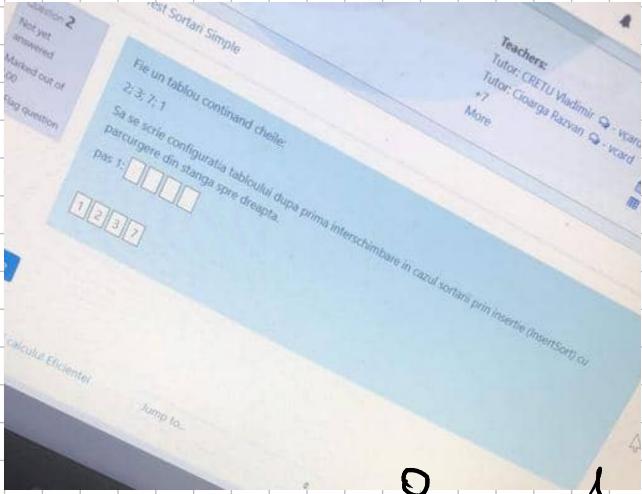
5	1	7	3
7	5	3	1



```

void insertion_sort(tip_element a[], int n)
{
    int i, j;
    tip_element tmp;
    for (i = 1; i < n; i++)
    {
        tmp = a[i];
        for (j = i; (j > 0) && (tmp.cheie < a[j - 1].cheie); j--)
            a[j] = a[j - 1];
        a[j] = tmp;
    }
}

```



initial

0	1	2	3
2	3	7	1

I. $\text{tmp} = a[1] = 3$ comparăm cu $a[0] = 2$

>

2	3	7	1
---	---	---	---

n. $\text{tmp} = a[2] = 7$ comparăm cu $a[1] = 3$

>

2	3	7	1
---	---	---	---

m. $\text{tmp} = a[3] = 1$ comparăm cu $a[2] = 7$

X

facem swap

→ 2 3 1 7

4; 14; 2; 6; 10

Sa se scrie configuratia tabloului dupa primele interschimbari in cazul sortarii prin insertie (InsertSort) cu parcurgere din stanga spre dreapta.

pas 1: 

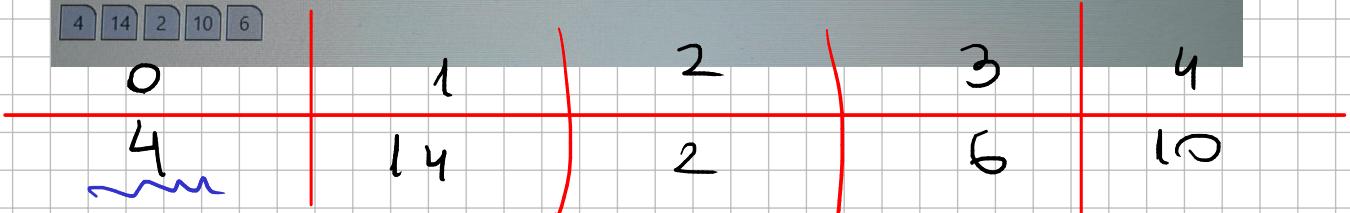
pas 2: 

pas 3: 

10 2 14 6 4

4 2 6 10 14

4 14 2 10 6



I. 14 comparare cu 4



$$i = 1$$

2 comparare cu 14



$$i = 2$$

swap

4 2 14 6 10

II. 2 comparare cu 4



swap

2 4 14 6 10

III. i = 3

6 comparare cu 14



swap

2 4 6 14 10

10 comparare cu 14

IV. i = 4

10 comparare cu 14



swap

2 4 6 10 14

comparare 10 cu 6 stop.

Fie un tablou continand cheile:

3; 1; 7; 2

Sa se scrie configuratia tabloului dupa prima interschimbare in cazul sortarii prin insertie (insertSort) cu parcurgere din stanga spre dreapta.

pas :

1	3	7	2
---	---	---	---

Fie un tablou continand cheile:

1; 4; 3; 2

Să se scrie configurația tabloului după primele interzchimbări în cazul sortării prin inserție (InsertSort) cu parcursare din stanga spre dreapta.

pas 1:

1	3	4	2
---	---	---	---

pas 2:

1	2	3	4
---	---	---	---

Fie un tablou continand cheile:

2; 3; 7; 1

Să se scrie configurația tabloului după prima interzchimbări în cazul sortării prin inserție (InsertSort) cu parcursare din stanga spre dreapta.

pas 1:

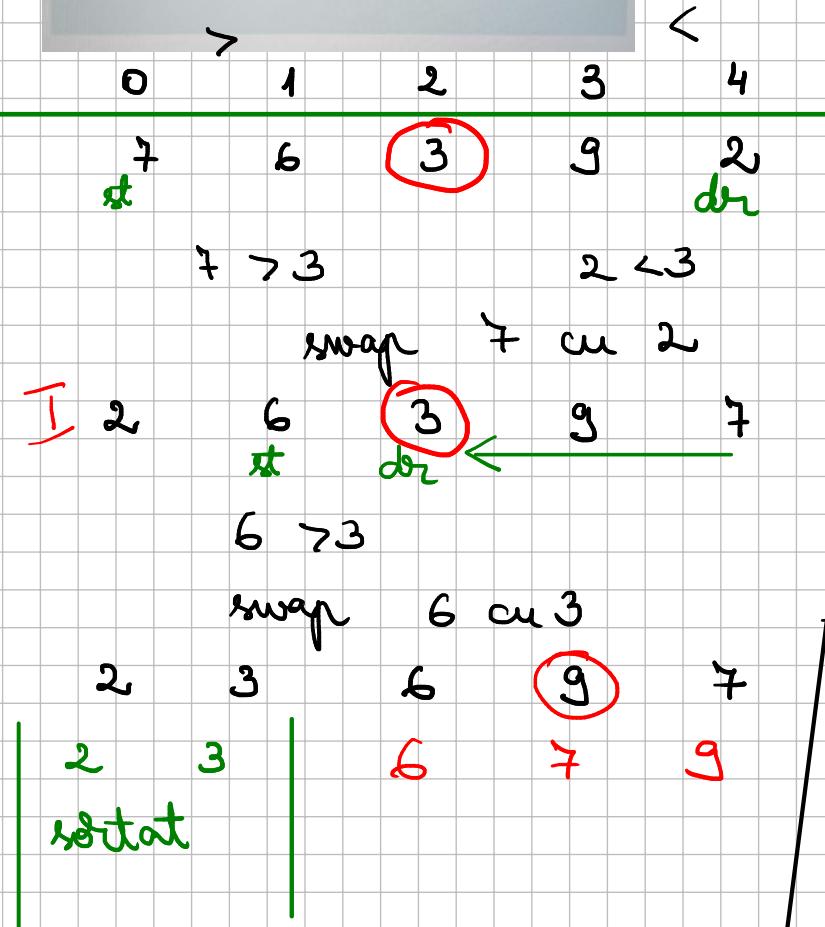
1	2	3	7
---	---	---	---

Sortari Avansate

Fie tabloul:
 $\{7; 6; 3; 9; 2\}$
 Sa se scrie configuratia tabloului dupa primii pasi ai sortarii prin partitionare (Quicksort), folosind pivot median, daca un pas este definit de exact o interschimbare:

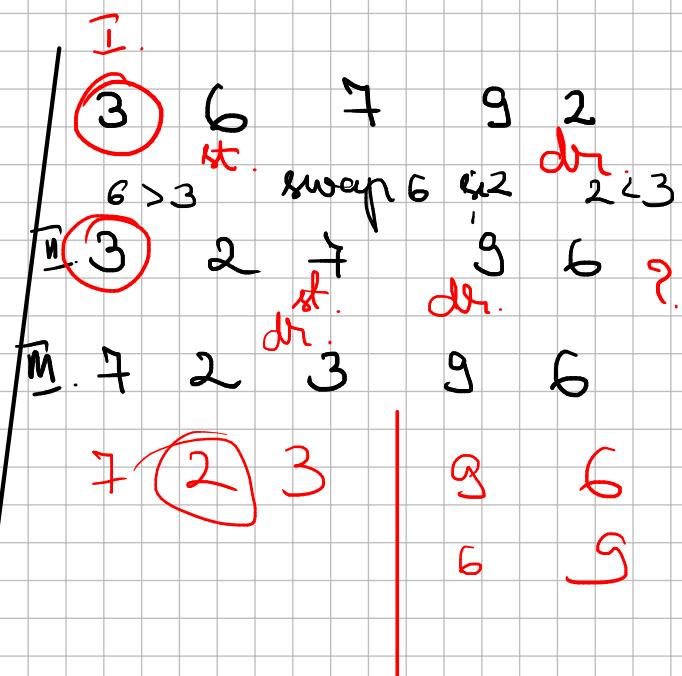
pasul 1: $\boxed{2} \boxed{6} \boxed{3} \boxed{9} \boxed{7}$
 pasul 2: $\boxed{2} \boxed{3} \boxed{6} \boxed{9} \boxed{7}$
 pasul 3: $\boxed{2} \boxed{3} \boxed{6} \boxed{7} \boxed{9}$

Timp ramas: 0:23:29



void quicksort(tip_element a[], int prim, int ultim)

```
{
    int stanga = prim + 1;
    int dreapta = ultim;
    //alegere pivot
    swap(&a[prim], &a[(prim + ultim) / 2]);
    //mutare pivot pe prima pozitie
    tip_element pivot = a[prim];
    while (stanga <= dreapta) //partitionare
    {
        while (a[stanga].cheie < pivot.cheie)
            stanga++;
        while (pivot.cheie < a[dreapta].cheie)
            dreapta--;
        if (stanga < dreapta)
            swap(&a[stanga++], &a[dreapta--]);
        else
            stanga++;
    }
    //mutare pivot la locul sau final
    swap(&a[dreapta], &a[prim]);
    //apelurile recursive
    if (prim < dreapta - 1)
        quicksort(a, prim, dreapta - 1);
    if (dreapta + 1 < ultim)
        quicksort(a, dreapta + 1, ultim);
}
```



Fie tabloul:

{1;2;4;3;5}

Sa se scrie configuratia tabloului dupa primii pasi ai sortarii in ordine **descrescatoare** folosind sortarea prin partitionare (Quicksort), folosind pivot median; daca un pas este definit de exact o interschimbare:

pasul 1:

4	2	3	1	5
---	---	---	---	---

pasul 2:

4	5	3	1	2
---	---	---	---	---

pasul 3:

5	4	3	2	1
---	---	---	---	---

0 1 2 3 4

1 2 **4**

I. 4 2 1 3 5

H. 4 5 **4** 3 2
4 5 1 3 2

M 1 5 4 3 2

0 1 2 3 4

~~st.~~ 1 2 **4**

I. 5 2 **4** 3 1
5 2 4 3 1

← dr.

dăm swap 4 cu 2 (indexuri menținute)

I 5 4 2 3 1

2 **3** 1
st. < dr. ←

swap. st. dr.

3 2 1

Rie urmatorul tablou:

$$m = 8$$

(9,2,5;1;4,7,3,8)

Sa se afiseze configuratia tabloului pentru pasul corespunzator incrementului cu valoarea 3, in cazul sortarii acestui tablou cu algoritmul shellsort.

H[i]=3:

9	2	5	1	4	7	3	8

$$R = 3$$

	0	1	2	3	4	5	6	7
	9			1			3	
		2			4			8
			5			7		
		1		3			9	
		2			4			8
			5			7		

$$R: \quad 1 \quad 2 \quad 5 \quad 3 \quad 4 \quad 7 \quad 9 \quad 8$$

Se considera un sistem incorporat cu resurse de memorie limitate in care folosirea recursivitatii sau a structurilor de tip stiva nu este permisa. Ce algoritm de sortare abt recomanda, in acest caz, pentru a avea o eficienta a timpului de rulare cat mai buna?

Selectati raspunsul corect:

- a. Sortarea prin metoda ansamblelor (Heap Sort) $\rightarrow O(n \log n)$ mediu
- b. Sortarea prin selectie performanta (Selection Sort) $\rightarrow O(n^2)$
- c. Sortarea prin insertie binara (Insert Sort) $\rightarrow O(n^2)$
- d. Sortarea prin partitionare (Quicksort) \rightarrow recursive

Question 1
Not yet
answered
Marked out of
1.00
Flag question

Fie un tablou continand cheile:
4; 1; 3; 7; 2
Sa se scrie configuratia tabloului dupa primele interschimbari in cazul sortarii prin insertie
(InsertSort) cu parcurgere din dreapta spre stanga.
pas 1:

--	--	--	--	--

pas 2:

--	--	--	--	--

pas 3:

--	--	--	--	--

3	1	7	2	4
---	---	---	---	---

3	4	7	2	1
---	---	---	---	---

4	2	7	3	1
---	---	---	---	---

9 1 3 7 2

I 7 comparare cu 2
 >

$$i = m - 2$$

swap

9 1 3 2 7

II. $i = m - 3$

3 comparare cu 2 >

swap

9 1 2 3 7

3 comp cu 7 < ✓

III. $i = m - 4$

1 comparare cu 2 < ✓

$i = m - 5$

9 comparare cu 1

swap

1 9 2 3 7

IV. 4 > 2 swap.

1 2 9 3 7

V. 9 > 3 swap

1 2 3 9 7

comp. cu 7 < ✓

Fie tabloul

(4;3;5;2;1)

Sa se scrie configuratia tabloului, in cazul sortarii cu radix direct, folosind o masca m pe 2 biti, in urmatoarele cazuri:

- dupa prima sortare folosind masca m

4	5	1	2	3
---	---	---	---	---

- dupa a doua sortare folosind masca m

1	2	3	4	5
---	---	---	---	---

direct \rightarrow dreapta la stanga

4	0	1	0	0
3	0	0	1	1
5	0	1	0	1
2	0	0	1	0
1	0	0	0	1

vector nr.

0	1	2	3
00	01	10	11
1	2	1	1
<hr/>			
1	3	4	5
<hr/>			
0	1	2	3
4	5	1	2
<hr/>			
1	2	3	4

vector nr.

4	0	1	0	0
5	0	1	0	1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

vector nr.

0	1	2	3
3	2	0	0
2	1	5	5
<hr/>			
0	1	2	3
1	2	3	4
<hr/>			
1	2	3	4

\rightarrow stabil, nu e in situ

Fie tabloul

{5;3;2;4;1}

Sa se scrie configuratia tabloului, in cazul sortarii cu radix direct, folosind o masca m pe 2 biti, in urmatoarele cazuri:

- dupa prima sortare folosind masca m

4 5 1 2 3

- dupa a doua sortare folosind masca m

1 2 3 4 5

Fie tabloul:

{7;9;3;2;4;6;8;5}

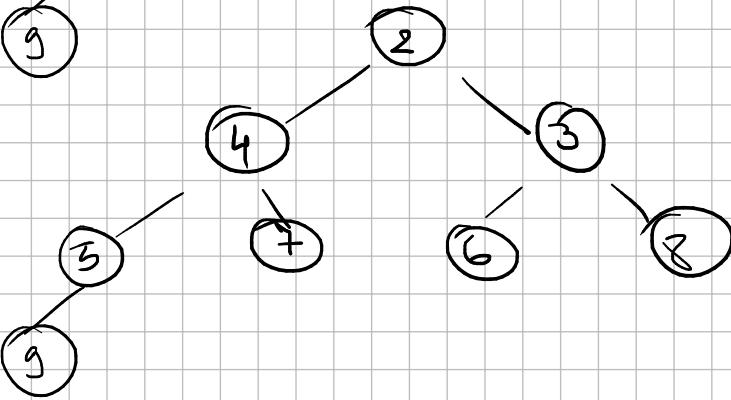
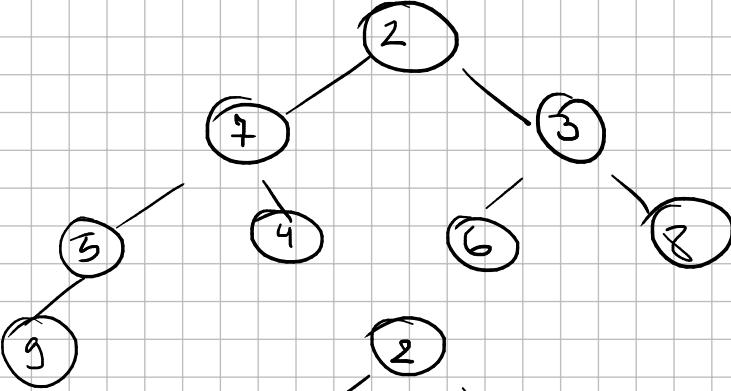
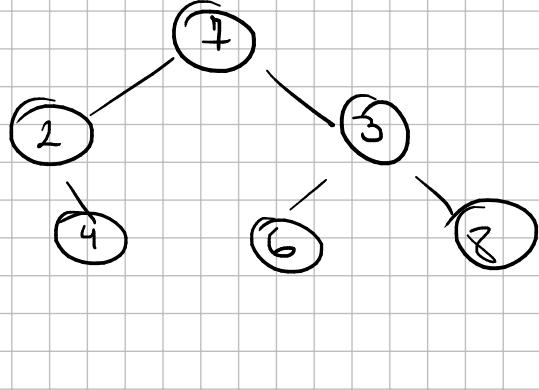
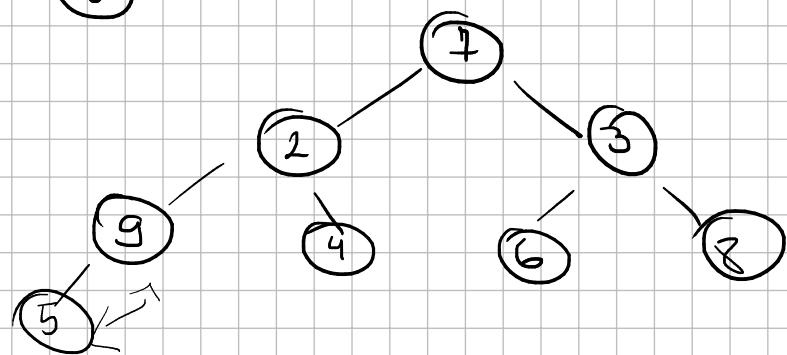
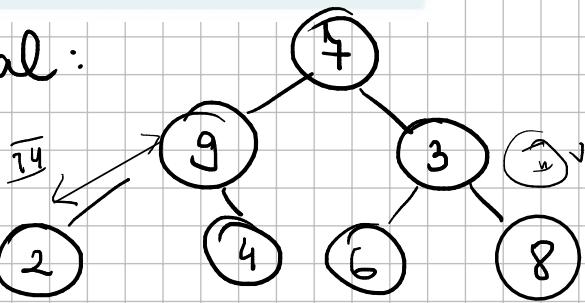
Sa se scrie configuratia tabloului, la pasul la care
ajunge pentru prima data ansamblu (min-heap).

Heap 1:

2	3	6	5	4	7	8	9
---	---	---	---	---	---	---	---

→ cel mai mic nr. sus

initial :



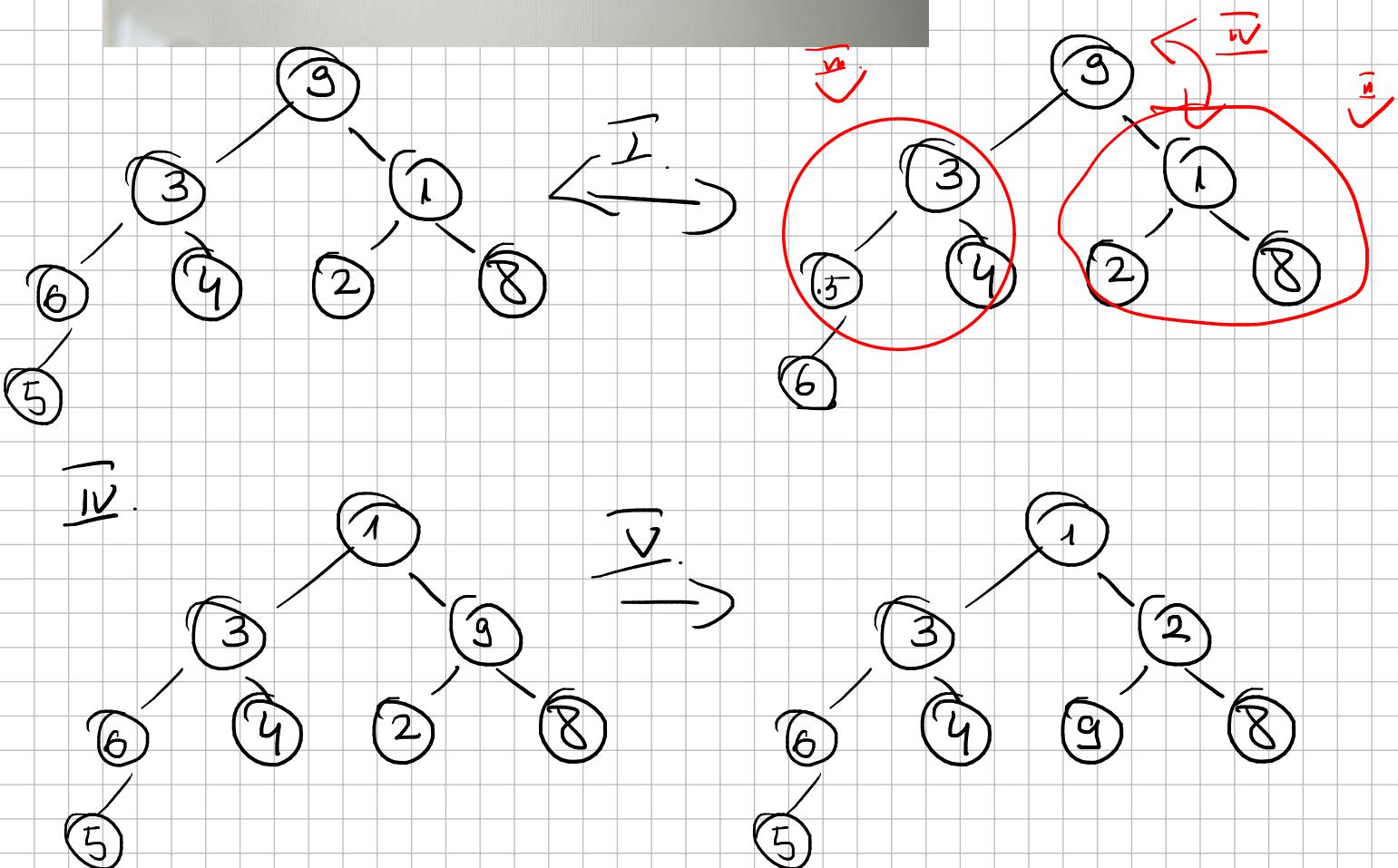
Fie tabloul:

{9;3;1;6;4;2;8;5}

Să se scrie configurația tabloului, la pasul la care ajunge pentru prima dată ansamblu (min-heap).

Heap 1:

1 4 2 5 6 3 8 9



R : 1 3 2 6 4 5 8 9

Fie urmatorul tablou:

(3;2;4;1;0)

Să se specifică configurația tabloului după primii pasi ai sortării binsort, considerând că un pas este definit de o interșimbare:

Pasul 1:

1	2	4	3	0
---	---	---	---	---

Pasul 2:

2	1	4	3	0
---	---	---	---	---

Pasul 3:

4	1	2	3	0
---	---	---	---	---

Pasul 4:

0	1	2	3	4
---	---	---	---	---

$\rightarrow O(n^2)$

0	1	2	3	4
3	2	4	1	0
<u>I</u>	swap {3,2} \rightarrow v[3]			
1	2	4	3	0
<u>II</u>	swap 1 \rightarrow v[1]			
2	1	4	3	0
<u>III</u>	swap 2 \rightarrow v[2]			
4	1	2	3	0
<u>IV</u>	swap 4 \rightarrow v[4]			
0	1	2	3	4

Corelații algoritmilor de sortare cu complexitatea lor(worst-case)

Radixsort

$O(nk)$



$O(n \cdot k)$

Bubblesort

$O(n^2)$



Heapsort

$O(n \log n)$



$O(n^2)$ (nu există)

$O(n \cdot (k / \text{sizeof(element)}))$

Sortarea "in situ" se realizează într-o zonă de memorie auxiliara celei alocate initial tabloului de sortat.

Selectați o opțiune:

Adevărat

Fals



Fie urmatorul tablou:

(1;3;2;4;0)

Sa se specifica configuratia tabloului dupa primii pasi ai sortarii binare, considerand ca un pas este definit de o interschimbare:

Pasul 1: 3 1 2 4 0

Pasul 2: 4 1 2 3 0

Pasul 3: 0 1 2 3 4

0 1 2 3 4

1 3 2 4 0

I. 3 1 2 4 0

n. 4 1 2 3 0

m. 0 1 2 3 4

Timp ramas 0:19:03

Se considera un sistem in care interschimbarile sunt foarte costisitoare. Ce algoritm de sortare ati recomanda pentru a minimiza numarul de interschimbari in general?

Selectati raspunsul corect:

- a. Sortarea prin metoda ansamblelor (Heap Sort) *multe*
 b. Sortarea prin interschimbare (Bubble Sort) $\rightarrow O(n^2)$
 c. Sortarea prin selectie performanta(Selection Sort)
 d. Sortarea prin insertie binara (Insert Sort)

Fie urmatorul tablou:

(7;9;3;2;4;6;8;5)

Sa se afiseze configuratia tabloului pentru pasul corespunzator incrementului cu valoarea 5, in cazul sortarii acestui tablou cu algoritmul shellsort:

$H[i]=5$:

6 8 3 2 4 7 9 5

0 1 2 3 4 5 6 7
7 9 3 2 4 6 8 5

7 9 3 2 4 6 8 5

2 4

R: 6 8 3 2 4 7 9 5

Fie urmatorul tablou:

(9;2;5;1;4;9;3;8)

Sa se afiseze configuratia tabloului pentru pasul corespunzator incrementului cu valoarea 5, in cazul sortarii acestui tablou cu algoritmul shellsort

H[0]=5:

7	2	5	1	4	9	3	8
---	---	---	---	---	---	---	---

$$n \lfloor \frac{i}{5} \rfloor = 5$$

0 1 2 3 4 5 6 7

9

2

5

3

8

1

4

7

9

2

3

5

8

1

4

R:

7 2 5 1 4] 9 3 8

Fie urmatorul tablou:

{9;2;5;1;4;7;3;8}

Sa se afiseze configuratia tabloului pentru pasul corespunzator incrementului cu valoarea 3, in cazul sortarii acestui tablou cu algoritmul shellsort

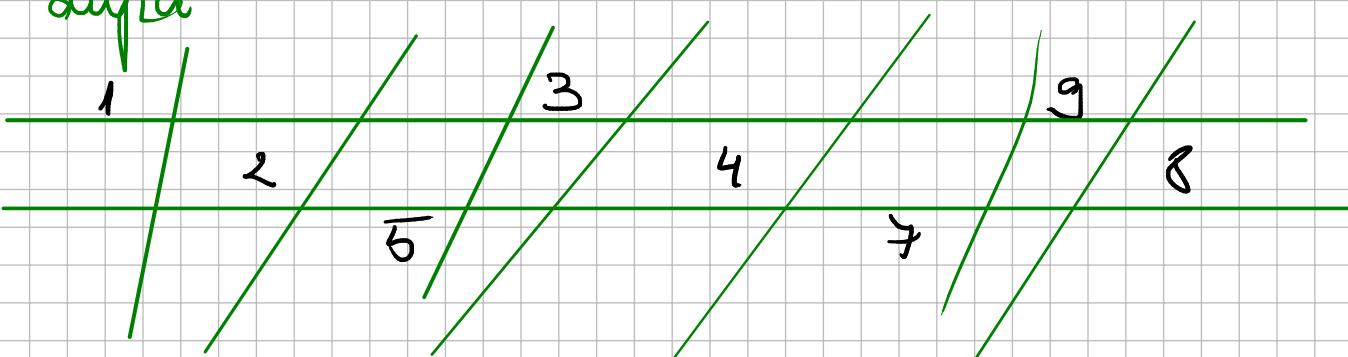
H[i]=3:

1	2	5	3	4	7	9	8
---	---	---	---	---	---	---	---

#C7 3

0	1	2	3	4	5	6	7
9		1			3		
	2			4			8
		5			7		

dupa



R: 1 2 5 3 4 7 9 8

Fie tabloul:

{7;6;3;9;2}

Sa se scrie configuratia tabloului dupa primii pasi ai sortarii prin partitionare (Quicksort), folosind pivot median, daca un pas este definit de exact o interschimbare:

pasul 1: 2 6 3 9 7

pasul 2: 2 3 6 9 7

pasul 3: 2 3 6 7 9

0	1	2	3	4
7	6	3	9	2
1	2	6	3	9
2	6	3	9	7
2	3	6	7	9

Cantitatea de memorie suplimentara alocata in cazul sortarii prin insertie depinde de dimensiunea tabloului de sortat.

in situ

Selectați o opțiune:

Adevărat

False

Fie tabloul:

{7;9;3;2;4}

Sa se scrie configuratia tabloului dupa primii pasi ai sortarii prin partitionare (Quicksort), folosind pivot median, daca un pas este definit de exact o interschimbare:

pasul 1: 

pasul 2: 

pasul 3: 







Fie tabloul

{4;5;3;1;2}

Sa se scrie configuratia tabloului, in cazul sortarii cu radix direct, folosind o masca m pe 2 biti, in urmatoarele cazuri:

- dupa prima sortare folosind masca m

- dupa a doua sortare folosind masca m

1 5 2 4 3

1 2 5 3 4

interschimbare

4	0	1	0	0
5	0	1	0	1
3	0	0	1	1
1	0	0	0	1
2	0	0	1	0

2
1
3
5
4

0	0	1	0
0	0	0	1
0	0	1	1
0	1	0	1
0	1	0	0

1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
5	0	1	0	1
4	0	1	0	0

1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1

Se considera un sistem care primeste date in mod dinamic (de exemplu de la retea/tastatura etc.). Ce metoda de sortare ati recomanda pentru a sorta elementele primite in ordine crescatoare?

Selectati raspunsul corect:

a. Sortarea binara (Binsort)

de la 0 la $n-1$ unice
luăm tot vector căutăm min

b. Sortarea prin selectie (Selection Sort)

c. Sortarea prin insertie (Insertion Sort)

d. Sortarea prin partitionare (Quick Sort)

luăm tot vector

Sterge alegerea mea

Fie tabloul

{1;3;5;2;7}

Sa se scrie configuratia tabloului, in cazul sortarii cu radix prin interschimbare, in urmatoarele cazuri:

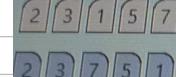
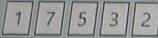
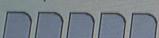
- dupa sortarea bitului 2 (celui mai semnificativ):



- dupa sortarea bitului 1:



- dupa sortarea bitului 0:



b₂

1	0	0	0	1
3	0	0	1	1
5	0	1	0	1
2	0	0	1	0
7	0	1	1	1

1	0	0	0	1
3	0	0	1	1
5	0	1	0	1
2	0	0	1	0
7	0	1	1	1

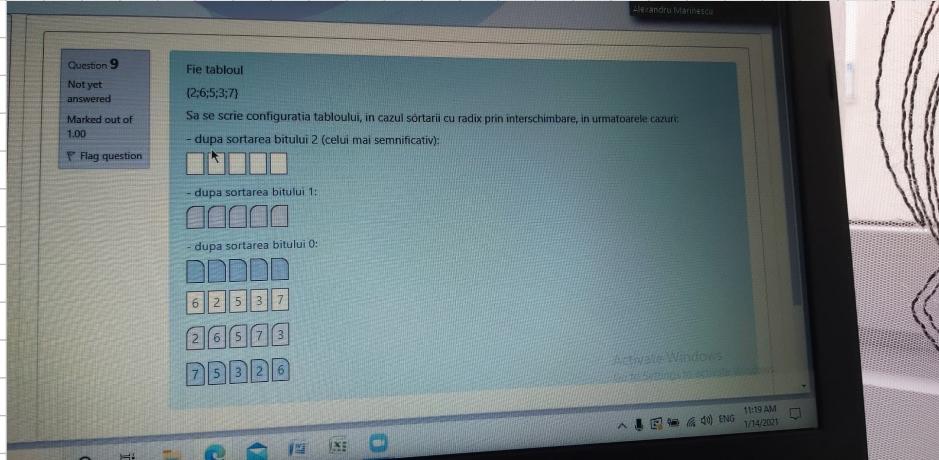
b₂

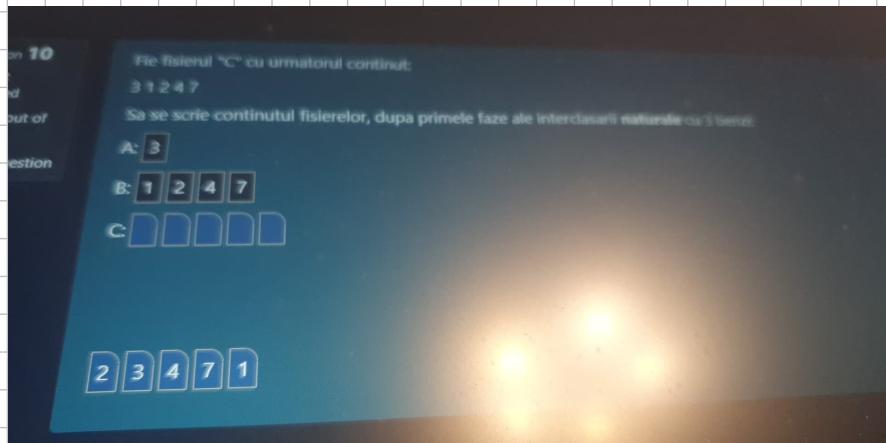
→

1	0	0	0	1
3	0	0	1	1
2	0	0	1	0
5	0	1	0	1
7	0	1	1	1

1	0	0	0	1
3	0	0	0	1
2	0	0	0	0
5	0	1	0	1
7	0	1	1	1

1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
5	0	1	0	1
7	0	1	1	1





3 → A

1 2 3 → B

2 → 1 → B

4 → 1 → B

7 → 1 → B

A: 3^{st.}

B: 1 2 4 7
 or

C: 1 2 3 4 7

Question 10
Not yet answered
Marked out of 1.00
Flag question

Ce este sortarea prin interclasare naturală?

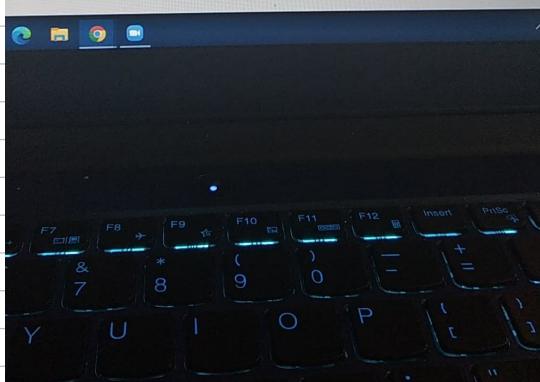
a. o sortare care interclasăaza monotonii
b. o sortare la care numarul de comparatii e foarte mic
c. e acea sortare la care la fiecare trecere lungimea monotonilor scade

Previous page

Ce e sortarea "in situ"?

a. o tehnica de sortare ce foloseste structuri de date suplimentare
b. o tehnica de sortare neperformanta
c. o tehnica de sortare ce foloseste chiar zona de memorie alocata taboului

Next page



Quiz navigation

Question 1
Not yet answered
Marked out of 1.00
Flag question

Când este indicat să folosim algoritmi de sortare avansati?

- a. cind dorim o performanta mai buna de timp
- b. toate variantele sunt corecte
- c. cand numarul elementelor ale tabloului de sortat este mare
- d. cand numarul elementelor ale tabloului de sortat este mai mic decat 100

Next page

Activate Windows
Go to Settings to activate Windows.

Time to search

accer

11:01 AM 14-Jan-2023

Question 2
Not yet answered
Marked out of 1.00
Flag question

File fisierul "C" cu urmatorul conținut:
7 3 5 9 15

Să se scrie continuul fisierelor, după primele fișiere ale intersecțiilor naturale (cu 3 benzi):

A:
B:
C:

9	7	5	3	15
3	7	9	15	5

Next page

You are screen sharing

11:01 AM 14-Jan-2023

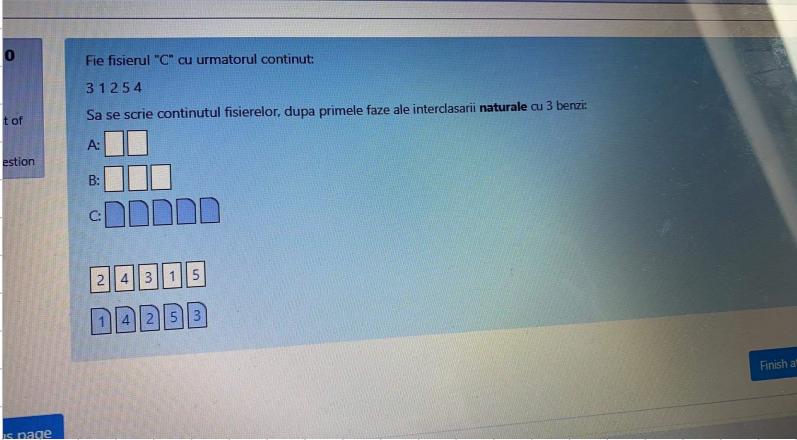
Question 2
Not yet answered
Marked out of 1.00
Flag question

Zicem că un algoritm de sortare al tablourilor este stabil dacă:

- a. dimensiunea tabloului nu se modifica;
- b. ordinea elementelor tabloului nu se schimbă;
- c. algoritmul nu foloseste memorie aditională;
- d. ordinea relativă a elementelor cu chei egale coincide cu cea initială;

Previous page

A : 7
B 3 5 9 15
C : 3 5 7 9 15



1 < 3 → B

2 > 1 → B

5 > 2 → B

4 < 5 → A

A : 3 4

B : 1 2 5

C : 1 2 3 4 5