

Structuri de date – Tema Seminar 1

1. Se dă un vector sortat. De câte ori apare x în vector?

Complexitatea funcției este $O(n)$, unde n este ultima poziție din vector a elementului căutat. Deoarece vectorul este sortat crescător sau descrescător, după ce am trecut de elementul căutat nu mai are rost să parcurgem elementele rămase, deci folosim break. Complexitatea algoritmului este $O(n \log n)$, aceasta fiind dată de sorted(vector).

Funcția de apariții putea fi implementată și folosind căutarea binară care are complexitate $O(\log n)$.

```
def aparitii (v, x):
    nr = 0
    for i in v:
        if i == x:
            nr += 1
        if ord == "C" or ord == "c":
            if i > x:
                break
        if ord == "D" or ord == "d":
            if i < x:
                break
    return nr

print()
print("Care este numarul de aparitii al unui element intr-un vector sortat?")
print("Inainte de a introduce elementele vectorului, precizati daca ordinea este crescatoare sau descrescatoare.")
print()
ord = input("Pentru ordinea crescatoare apasati tasta C, iar pentru ordinea descrescatoare apasati tasta D: ")
v = []

if ord == "C" or ord == "c" or ord == "D" or ord == "d":
    for i in input("Introduceti elementele vectorului: ").split():
        v.append(int(i))
else:
    print()
    print("Nu exista aceasta ordine! Rulati din nou programul si introduceti datele corect!")

if ord == "C" or ord == "c":
    sc = sorted(v)
    if v != sc:
        print()
        print("Elementele introduse nu sunt in ordine crescatoare! Rulati din nou programul si introduceti datele corect!")
    else:
        val = int(input("Introduceti valoarea elementului al carui numar de aparitii doriti sa il aflati: "))
        print()
```

```

        print(f"Elementul {val} apare de {aparitii(v, val)} ori.")

if ord == "D" or ord == "d":
    sd = sorted(v, reverse=True)
    if v != sd:
        print()
        print("Elementele introduse nu sunt in ordine descrescatoare! Rulati
din nou programul si introduceti datele corect!")
    else:
        val = int(input("Introduceti valoarea elementului al carui numar de
aparitii doriti sa il aflatii: "))
        print()
        print(f"Elementul {val} apare de {aparitii(v, val)} ori.")

```

Rularea programului:

```

Care este numarul de aparitii al unui element intr-un vector sortat?
Inainte de a introduce elementele vectorului, precizati daca ordinea este crescatoare sau descrescatoare.

Pentru ordinea crescatoare apasati tasta C, iar pentru ordinea descrescatoare apasati tasta D: C
Introduceti elementele vectorului: 1 2 5 77 77 77 77 77 89 100 121 178 209 209 209 567 589
Introduceti valoarea elementului al carui numar de aparitii doriti sa il aflatii: 77

Elementul 77 apare de 5 ori.

```

```

Care este numarul de aparitii al unui element intr-un vector sortat?
Inainte de a introduce elementele vectorului, precizati daca ordinea este crescatoare sau descrescatoare.

Pentru ordinea crescatoare apasati tasta C, iar pentru ordinea descrescatoare apasati tasta D: D
Introduceti elementele vectorului: 900 876 876 876 84 43 22 1
Introduceti valoarea elementului al carui numar de aparitii doriti sa il aflatii: 876

Elementul 876 apare de 3 ori.

```

2. Care este numărul de inversiuni într-un vector?

Fie v un vector cu elementele x_1, x_2, \dots, x_n . Parcurgând vectorul cu doi indici, cuprinși între 1 și n , $i < j$, dacă $x_i > x_j$ avem o inversiune.

```

def sortare (v):
    nr = 0
    for i in range(len(v)-1):
        for j in range(i+1, len(v)):
            if v[i] > v[j]:
                nr += 1
                inv.append((v[i], v[j]))
    return nr

print()
print("Care este numarul de inversiuni intr-un vector?")

```

```

v = []
inv = []
for i in input("Introduceti elementele vectorului: ").split():
    v.append(int(i))

inversiuni = sortare(v)

print()
print(f"Vectorul are {inversiuni} inversiuni, acestea fiind:")
for inversiune in inv:
    print(inversiune)

```

Rularea programului:

```

Care este numarul de inversiuni intr-un vector?
Introduceti elementele vectorului: 4 6 4 2 9 8 10

Vectorul are 5 inversiuni, acestea fiind:
(4, 2)
(6, 4)
(6, 2)
(4, 2)
(9, 8)

```

Ajutându-ne de inversiuni putem sorta un vector prin interclasare. Prin modificările făcute pe parcurs, observăm că numărul de inversiuni se modifică – scade.

```

def sortare (v):
    nr = 0
    for i in range(len(v)-1):
        for j in range(i+1, len(v)):
            if v[i] > v[j]:
                aux = v[i]
                v[i] = v[j]
                v[j] = aux
                nr += 1
                inv.append((v[j], v[i]))
            # deoarece am inversat cele doua elemente, trebuie sa punem
invers valorile in tuplu
    return nr

print()
print("Care este numarul de inversiuni intr-un vector?")

v = []
inv = []
for i in input("Introduceti elementele vectorului: ").split():
    v.append(int(i))

inversiuni = sortare(v)

```

```
print()
print(f"Vectorul a fost sortat prin interclasare: ")
for i in v:
    print(i, end=' ')
print()
print(f"Vectorul are {inversiuni} inversiuni, acestea fiind:")
for inversiune in inv:
    print(inversiune)
```

Rularea programului:

```
Care este numarul de inversiuni intr-un vector?
Introduceti elementele vectorului: 4 6 4 2 9 8 10

Vectorul a fost sortat prin interclasare:
2 4 4 6 8 9 10
Vectorul are 4 inversiuni, acestea fiind:
(4, 2)
(6, 4)
(6, 4)
(9, 8)
```