FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 17 · LECȚIA PY9-17

Metode de sortare: selecție, frecvențe, bule

Durată: 150 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești pas cu pas selecția minimului și metoda bulelor și recunoști o metodă după urmele ei.
- Numeri comparațiile și interschimbările și alegi metoda potrivită datelor.
- Adaptezi metodele: descrescător, după un criteriu, doar pe o parte a listei, pe liste paralele.

Ce trebuie să știi (o pagină)

```
def selectie(a):                        # selecția minimului
    n = len(a)
    for i in range(n - 1):
        p = i
        for j in range(i + 1, n):
            if a[j] < a[p]:
                p = j
        a[i], a[p] = a[p], a[i]

def bule(a):                            # metoda bulelor
    n = len(a)
    for k in range(n - 1):
        schimbat = False
        for j in range(n - 1 - k):
            if a[j] > a[j + 1]:
                a[j], a[j + 1] = a[j + 1], a[j]
                schimbat = True
        if not schimbat:
            break
```

	Selecția minimului	**Bulele**	**Listă de frecvențe (valori `0` … `M`)**
ideea	pune minimul rămas pe locul `i`	schimbă vecinii în ordine greșită	numără, apoi rescrie în ordinea valorilor
după pasul `k`	primele `k` sunt la locul final	ultimele `k` sunt la locul final	—
comparații	mereu `n(n − 1)/2`: $O(n^2)$	`n − 1` … `n(n − 1)/2`: $O(n)$ … $O(n^2)$	niciuna: $O(n + M)$
interschimbări	cel mult `n − 1`	= perechile inversate	niciuna
descrescător	caută maximul (`>`)	`<` în loc de `>`	parcurgi valorile de la `M` la 0

Nivelul 1 · Prezice

★ citește și prezice

P1. Cum arată `[5, 2, 8, 1, 9, 3]` după primul, al doilea și al treilea pas al selecției minimului?

P2. Cum arată aceeași listă `[5, 2, 8, 1, 9, 3]` după prima și după a doua trecere a metodei bulelor?

P3. Ce afișează?

```
a = [3, 1, 2]
comp = sw = 0
n = len(a)
for k in range(n - 1):
    for j in range(n - 1 - k):
        comp += 1
        if a[j] > a[j + 1]:
            a[j], a[j + 1] = a[j + 1], a[j]
            sw += 1
print(a, comp, sw)
```

P4. Ce afișează? Ce reprezintă rezultatul pentru numărul `n`?

```
n = 30512
fr = [0] * 10
for c in str(n):
    fr[int(c)] += 1
rez = ""
for cif in range(9, -1, -1):
    rez += str(cif) * fr[cif]
print(rez)
```

P5. Ce afișează?

```
a = [3, -7, 5, -1]
b = sorted(a, key=abs)
a.sort(reverse=True)
print(a, b)
```

P6. Ce afișează fiecare `print(a)` și ce valoare are `sw` la final?

```
a = [4, 9, 2, 7]
sw = 0
for i in range(len(a) - 1):
    p = i
    for j in range(i + 1, len(a)):
        if a[j] > a[p]:
            p = j
    if p != i:
        a[i], a[p] = a[p], a[i]
        sw += 1
    print(a)
print(sw)
```

P7. Pentru `[1, 2, 3, 5, 4]`, câte treceri și câte comparații face funcția `bule` din chenarul de pe prima pagină?

P8. Ghicește metoda. Lista `[4, 3, 1, 2]` a fost sortată de două ori, cu metode diferite. Iată starea după fiecare pas. Care este selecția și care sunt bulele? Cum îți dai seama?

```
Metoda X:  [1, 3, 4, 2]   [1, 2, 4, 3]   [1, 2, 3, 4]
Metoda Y:  [3, 1, 2, 4]   [1, 2, 3, 4]   [1, 2, 3, 4]
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Varianta de selecție de mai jos a trecut testul `[3, 1, 2]` → `[1, 2, 3]`. Pentru lista deja sortată `[1, 2, 3]` dă însă `[2, 1, 3]`. Unde e greșeala? Ce lecție despre testare scoți de aici?

```
n = len(a)
p = 0
for i in range(n - 1):
    for j in range(i + 1, n):
        if a[j] < a[p]:
            p = j
    a[i], a[p] = a[p], a[i]
```

P10. Pe lista sortată `[1, 2, 3, 4]` programul face corect o singură trecere (afișează `[1, 2, 3, 4]` `1`). Pe `[2, 1, 3, 4]` face însă 3 treceri în loc de 2. Ce rând este pus greșit?

```
n = len(a)
treceri = 0
schimbat = False
for k in range(n - 1):
    treceri += 1
    for j in range(n - 1 - k):
        if a[j] > a[j + 1]:
            a[j], a[j + 1] = a[j + 1], a[j]
            schimbat = True
    if not schimbat:
        break
print(a, treceri)
```

P11. Sortarea cu frecvențe pentru `note = [3, 1, 3]` afișează `[1, 2, 2]`. Ce s-a adăugat în `rez`, de fapt? Repară.

```
fr = [0] * 11
for x in note:
    fr[x] += 1
rez = []
for v in range(1, 11):
    for _ in range(fr[v]):
        rez.append(fr[v])
print(rez)
```

P12. Programul trebuie să afișeze **cele mai mari 3** punctaje, descrescător. Pentru `p = [56, 91, 78, 100, 64, 91]` afișează `[56, 64, 78]`. Repară-l în două feluri.

```
print(sorted(p)[:3])
```

P13. *Rânduri amestecate.* Programul sortează **descrescător** prin selecție (a maximului). Un rând este în plus.

```
A)      a[i], a[p] = a[p], a[i]
B)  for i in range(n - 1):
C)          if a[j] > a[p]:
D)      p = i
E)  a = list(map(int, input().split())); n = len(a)
F)              p = j
G)      for j in range(i + 1, n):
H)          if a[j] < a[p]:
I)  print(*a)
```

Ce afișează pentru `4 9 2 7 9`?

P14. *Din pseudocod de BAC în Python.* Algoritmul repetă trecerile **până când** nu mai există interschimbări:

```
repetă
    ok ← adevărat
    pentru i ← 0, n − 2 execută
        dacă a[i] > a[i+1] atunci  a[i] ↔ a[i+1];  ok ← fals
până când ok
```

Traducerea de mai jos lasă `[3, 1, 2]` nesortată. De ce? Scrie traducerea corectă.

```python
ok = False
while ok:
    ok = True
    for i in range(len(a) - 1):
        if a[i] > a[i + 1]:
            a[i], a[i + 1] = a[i + 1], a[i]
            ok = False
```

Nivelul 3 · Scrie cod

★★ scrie cod

Fără `sort()` sau `sorted()` la P15–P20. Verifică **toate** testele înainte să spui că e gata și adaugă-le pe cele obligatorii la orice sortare: listă vidă, un element, listă deja sortată, listă inversă, valori egale.

P15. Temperaturi, descrescător. Sortează descrescător cu metoda bulelor (cu `schimbat`) și afișează lista, apoi numărul de interschimbări.

intrare	ieșire
`3 -1 7 0`	`7 3 0 -1` / `3`
`9 5 1`	`9 5 1` / `0`
`1 2 3 4`	`4 3 2 1` / `6`
`4 4 2`	`4 4 2` / `0`

P16. Secvențele pozitive. Sortează crescător, cu metoda bulelor, **fiecare secvență maximă de numere pozitive** (`> 0`) aflate pe poziții consecutive; celelalte numere rămân pe loc. Scrie o funcție

`bule(a, st, dr)` care sortează doar `a[st..dr]`.

intrare	ieșire
`3 1 -2 5 4 9 0 2 1`	`1 3 -2 4 5 9 0 1 2`
`-1 0 -5`	`-1 0 -5`
`4 3 2 1`	`1 2 3 4`
`2 -1 1`	`2 -1 1`
`0 9 8`	`0 8 9`

P17. Notele la informatică. Notele sunt întregi de la 1 la 10. Sortează-le **descrescător** cu listă de frecvențe, apoi afișează nota maximă și câți elevi au luat-o.

intrare	ieșire
`7 10 4 10 9`	`10 10 9 7 4` / `10 2`
`5`	`5` / `5 1`
`1 1 1`	`1 1 1` / `1 3`
`3 8 6 8 2 6 8`	`8 8 8 6 6 3 2` / `8 3`

P18. Comenzile online, pas cu pas. Citește valorile unor comenzi și sortează-le crescător prin selecția minimului, afișând lista **după fiecare pas**.

intrare	ieșire
`3 1`	`1 3`
`8`	(nimic)
`120 45 300 80 45 210 99`	6 rânduri; primul `45 120 300 80 45 210 99`, al doilea `45 45 300 80 120 210 99`, ultimele trei `45 45 80 99 120 210 300`

P19. Cererile de mentenanță. O firmă de lifturi primește `n` cereri, fiecare cu un identificator și un cod de defecțiune de la 1 la 8. Afișează identificatorii ordonați după cod; la cod egal, păstrezi ordinea sosirii. Folosește 9 liste (una pentru fiecare cod), ca o listă de frecvențe care ține minte **cine**, nu doar **câți**.

intrare	ieșire
`5` / `L7 3` / `L2 1` / `L9 3` / `L4 8` / `L1 1`	`L2 L1 L7 L9 L4`
`1` / `A 5`	`A`
`3` / `X 8` / `Y 8` / `Z 8`	`X Y Z`

P20. După suma cifrelor. Sortează numerele crescător după suma cifrelor; la sumă egală, numărul mai mic întâi. Adaptează selecția minimului: scrie o funcție `inainte(x, y)` care spune dacă `x` trebuie pus înaintea lui `y` și folosește-o în locul lui `<`.

intrare	ieșire
`91 12 300 45 7`	`12 300 7 45 91`
`19 28 37`	`19 28 37`
`50 5 14`	`5 14 50`
`0 10`	`0 10`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Comparații la olimpiadă. Sunt 200 de punctaje. (a) Câte comparații face selecția pentru sortarea completă? Depinde de ordinea inițială? (b) Câte comparații fac selecția și bulele dacă vrei doar primele 10 punctaje (te oprești după 10 pași)? (c) Câte comparații fac bulele cu `schimbat`, în cel mai bun caz, pe 200 de valori?

P22. Cronometrul. Generează 4000 de numere aleatoare între 0 și 100 (`random.randint`) și măsoară cu `time.perf_counter()` timpul pentru: selecție, bule, sortarea cu frecvențe și `sorted()`. Repetă pentru o listă sortată în care doar două elemente vecine sunt inversate. Ce clasament obții în fiecare caz? Explică-l.

P23. Problemă inversă. Găsește o permutare a numerelor `1, 2, 3, 4, 5` pentru care selecția minimului face **o singură** interschimbare utilă (cu `p != i`), iar metoda bulelor are nevoie de **toate** cele 4 treceri. Câte interschimbări fac bulele pe permutarea găsită?

P24. De ce funcționează bulele? (a) Explică de ce, după trecerea `k`, ultimele `k` elemente sunt la locul lor final (de aici `range(n - 1 - k)`). (b) Îmbunătățire: ține minte poziția `ultim` a **ultimei** interschimbări dintr-o trecere; tot ce e după ea este deja sortat, deci trecerea următoare se poate opri acolo. Implementează și compară numărul de comparații cu varianta cu `schimbat` pe `[2, 1, 3, 4, 5, 6, 7, 8]` și pe `[3, 1, 2, 4, 5, 6, 7, 8, 9, 10]`.

P25. Mini-proiect: programările clinicii. Ai două liste paralele: `nume` și `durata` (minute). Ordonează programările crescător după durată cu metoda bulelor, mutând **împreună** numele și durata, și afișează starea după fiecare trecere. Apoi afișează programul zilei, pornind de la ora 9:00. Pentru `nume = ["Pop", "Ene", "Dinu", "Lupu"]` și `durata = [30, 15, 45, 15]`:

```
trecerea 1: Ene(15) Pop(30) Lupu(15) Dinu(45)
trecerea 2: Ene(15) Lupu(15) Pop(30) Dinu(45)
trecerea 3: Ene(15) Lupu(15) Pop(30) Dinu(45)
09:00 Ene
09:15 Lupu
09:30 Pop
10:00 Dinu
```

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] #509 **ordonare** — ușoară
  https://www.pbinfo.ro/probleme/509/ordonare

- [ ] #244 **cifreord** — ușoară
  https://www.pbinfo.ro/probleme/244/cifreord
- [ ] #129 **sortare** — ușoară
  https://www.pbinfo.ro/probleme/129/sortare

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #130 **sortcif** — ușoară
  https://www.pbinfo.ro/probleme/130/sortcif
- [ ] #511 **ksort** — ușoară
  https://www.pbinfo.ro/probleme/511/ksort
- [ ] #183 **sortpp** — medie
  https://www.pbinfo.ro/probleme/183/sortpp
