FIȘĂ DE LUCRU · CLASA A X-A · FIȘA 6 · ACOPERĂ LECȚIA X.7

Secvențe și sume parțiale

Durată: 90 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Găsești cea mai lungă secvență cu o proprietate și o numeri, într-o singură parcurgere.
- Răspunzi în O(1) la întrebări de sumă pe interval cu sume parțiale.
- Aplici Kadane și fereastra glisantă și știi de ce funcționează.

Ce trebuie să știi (o pagină)

Ce vrei	Cod	Cost
secvență = elemente **consecutive**; subșir = aceeași ordine, nu neapărat vecine		
un vector cu `n` elemente are `n(n+1)/2` secvențe		toate secvențele: $O(n^2)$
cea mai lungă secvență cu proprietatea P	`if (P) lung++; else lung = 0;` `if (lung > lmax) { lmax = lung; st = i - lung + 1; }`	O(n)
proprietate care depinde de vecin (crescătoare, egale)	`if (a[i] > a[i-1]) lung++; else lung = 1;`	elementul curent începe secvența nouă
sume parțiale	`s[0] = 0; s[i] = s[i-1] + a[i];`	O(n) o dată
suma `a[x..y]`	`s[y] - s[x-1]`	O(1) pe întrebare
secvența de sumă maximă (Kadane)	`if (sc < 0) { sc = 0; st = i; }` `sc += a[i];` `if (sc > smax) …`	O(n), `smax = a[1]`
fereastră de lungime fixă `k`	`s += a[i] - a[i-k];`	O(n)

ATENȚIE — GREȘELI FRECVENTE

Sumele pot depăși `int` (10^5 valori de 10^9): folosește `long long` pentru `s[]`, `sc` și `smax`. Și nu uita: `s[0] = 0`, altfel întrebările cu `x = 1` ies greșit.

Nivelul 1 · Prezice

★ citește și prezice

P1. Vectorul `a = (4, 7, 1, 7, 9)`. Pentru fiecare șir, spune dacă este **secvență**, doar **subșir** sau **niciuna**: `(7, 1, 7)`, `(4, 1, 9)`, `(7, 4)`, `(1, 7, 9)`, `(4, 7, 1, 7, 9)`. Câte secvențe are `a` în total? Câte au lungimea 3?

P2. Completează sumele parțiale, apoi calculează **doar din tabel** sumele `a[2..5]`, `a[1..6]` și `a[4..4]`.

i	0	1	2	3	4	5	6
a[i]	–	2	−1	3	0	5	−4
s[i]	0						

P3. Problemă inversă. Cineva a calculat sumele parțiale `s = (0, 3, 3, 8, 6, 10)` (de la `s[0]` la `s[5]`) și a pierdut vectorul. Reconstruiește `a[1..5]`. Ce formulă ai folosit?

P4. Pseudocod.

```
citește n, x
p ← x; l ← 1; lmax ← 1
┌pentru i ← 2,n execută
│ citește x
│┌dacă x = p atunci l ← l + 1
│├altfel l ← 1
│└■
│┌dacă l > lmax atunci lmax ← l
│└■
│ p ← x
└■
scrie lmax
```

a) Ce se afișează pentru `n = 9` și `5 5 2 2 2 7 2 2 2`? **b)** Pentru `n = 6`, dă un șir pentru care se afișează `1` și unul pentru care se afișează `6`. **c)** Ce se afișează pentru șirul de la a) dacă atribuirea `p ← x` se mută imediat după `citește x`?

P5. Urmărește Kadane (codul din lecție, `smax = a[1]`) pe `a = (−2, 3, −1, −4, 5, −1, 2, −3)`. Marchează pașii la care se face reset.

i	1	2	3	4	5	6	7	8
sc (după)								
smax								
st								

P6. Ce afișează pentru `n = 6`, `k = 3`, `a = (4, 1, 3, 6, 2, 5)`? Ce înseamnă numerele afișate?

```cpp
long long s = 0;
for (int i = 1; i <= k; i++) s += a[i];
cout << s;
for (int i = k + 1; i <= n; i++)
{
    s += a[i] - a[i - k];
    cout << ' ' << s;
}
```

P7. Ce afișează pentru `a = (1, 3, 5, 2, 4, 4, 6, 7)`? Spune în cuvinte ce numără.

```cpp
int cnt = 1;
for (int i = 2; i <= n; i++)
    if (a[i] <= a[i - 1]) cnt++;
cout << cnt;
```

P8. Pentru `a = (2, 3, 0, 5, −5, 5)`: ce afișează secvența și de câte ori se execută `s += a[j]`? Leagă al doilea număr de formula `n(n+1)/2`.

```
int cnt = 0;
for (int i = 1; i <= n; i++)
{
    int s = 0;
    for (int j = i; j <= n; j++)
    {
        s += a[j];
        if (s == 5) cnt++;
    }
}
cout << cnt;
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Elevul răspunde la întrebarea „suma `a[x..y]`" cu `s[y] - s[x]`. Ce obține pentru `a = (3, 1, 4, 1, 5)`, `x = 2`, `y = 4`? Ce element lipsește mereu din sumă? Repară.

P10. Kadane cu poziții, scris „altfel". Suma afișată e corectă, dar poziția de început nu. Ce afișează pentru `a = (2, −5, 3, 4)`? Repară **un singur** rând.

```
for (int i = 1; i <= n; i++)
{
    sc += a[i];
    if (sc > smax) { smax = sc; bestSt = st; bestDr = i; }
    if (sc < 0) { sc = 0; st = i; }
}
cout << smax << ' ' << bestSt << ' ' << bestDr;
```

P11. Cea mai lungă secvență **strict crescătoare** (poziția de început și lungimea). Ce afișează pentru `5 1 2 3`? Care e răspunsul corect? Repară.

```
int lung = 1, lmax = 1, st = 1;
for (int i = 2; i <= n; i++)
{
    if (a[i] > a[i - 1]) lung++;
    else lung = 0;
    if (lung > lmax) { lmax = lung; st = i - lung + 1; }
}
cout << st << ' ' << lmax;
```

P12. *Puzzle cu rânduri amestecate.* Programul afișează poziția de început a primei ferestre de `k` elemente cu suma maximă (vectorul e deja citit). Pune rândurile în ordine; unul este în plus.

```
A)  for (int i = k + 1; i <= n; i++) {
B)  long long s = 0, best;  int poz = 1;
C)      s += a[i] - a[i - k];
D)  for (int i = 1; i <= k; i++) s += a[i];
E)      if (s > best) { best = s; poz = i - k + 1; }
F)  best = s;
G)  }
H)      s += a[i];
I)  cout << poz;
```

P13. Completează golurile ca programul să răspundă la `m` întrebări „câte numere **pare** sunt între pozițiile `x` și `y`?". Test: `a = (4, 7, 2, 8, 1, 6)`, întrebările `(1, 6)`, `(2, 3)`, `(5, 5)` → `4 1 0`.

```
for (int i = 1; i <= n; i++)
{
    cin >> a[i];
    p[i] = p[i - 1] + (_________________);
}
// pentru fiecare întrebare (x, y):
cout << _________________ << ' ';
```

P14. Ce greșeală a făcut elevul? „Secvența de sumă maximă se obține adunând toate elementele pozitive." Arată pe `(3, −10, 4)` și pe `(5, −1, 5)` că răspunsul lui e greșit. Ce a confundat?

Nivelul 3 · Scrie cod

★★ scrie cod

P15. Seria fără înfrângere. Se dau diferențele de goluri din `n` meciuri ale unei echipe (pozitiv = victorie, 0 = egal, negativ = înfrângere). Afișează lungimea celei mai lungi serii de meciuri consecutive **fără înfrângere** și meciul cu care începe (prima, la egalitate); dacă nu există, `0 0`.

intrare	ieșire
`8` / `2 0 -1 1 1 0 3 -2`	`4 4`
`3` / `-1 -2 -1`	`0 0`
`4` / `1 1 1 1`	`4 1`

P16. Traseul pe munte. Se dau altitudinile a `n` puncte de pe un traseu și `q` întrebări `(x, y)`, cu `x ≤ y`: câți metri **urci** în total mergând de la punctul `x` la punctul `y`? (Coborârile nu se scad.) Fiecare răspuns în O(1). Atenție la ce scazi!

intrare	ieșire
`6` / `1000 1200 1100 1500 1400 1600` / `3` / `1 6` / `2 4` / `3 3`	`800` / `400` / `0`

P17. Cea mai bună perioadă a magazinului. Se dă profitul pe `n` zile (poate fi negativ). Afișează suma maximă a unei perioade de zile consecutive, plus ziua de început și de sfârșit. Diferență față de lecție: dacă toate zilele sunt pe pierdere, magazinul preferă să stea închis: afișează doar `0`.

intrare	ieșire
`7` / `-2 5 -1 4 -8 3 2`	`8 2 4`
`3` / `-3 -1 -2`	`0`
`1` / `6`	`6 1 1`

P18. Concediul. Prețurile unei camere de hotel pe `n` nopți. Vrei exact `k` nopți consecutive, cât mai ieftin. Afișează prima noapte și costul total (prima variantă, la egalitate).

intrare	ieșire
`7 3` / `120 90 100 80 150 70 60`	`2 270`
`4 4` / `10 20 30 40`	`1 100`
`5 1` / `50 40 40 60 30`	`5 30`

P19. Zilele însorite. Pentru `n` zile se dă `1` (soare) sau `0` (nori). Câte **perioade** de zile consecutive au fost în întregime însorite? (De exemplu, 3 zile însorite la rând conțin 6 perioade.) O singură parcurgere, fără vector.

intrare	ieșire
`4` / `1 1 0 1`	`4`
`2` / `0 0`	`0`
`3` / `1 1 1`	`6`

P20. Ciocolata împărțită corect. O tabletă are `n` pătrățele cu numere de alune `a[1..n]`. Găsește prima poziție `p` (`1 ≤ p < n`) la care, rupând după pătrățelul `p`, ambele bucăți au același număr de alune; altfel `nu`.

intrare	ieșire
`6` / `3 1 2 4 1 1`	`3`
`3` / `1 1 1`	`nu`
`2` / `5 5`	`1`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Construiește vectorul. Dă un vector cu 6 elemente **nenule** pentru care secvența de sumă maximă are suma 10, începe la poziția 3 și se termină la 5, iar suma tuturor elementelor este **negativă**. Verifică-l cu Kadane.

P22. Vectorul de diferențe. Un vector de 5 zerouri primește `m` operații „adaugă `v` la `a[x..y]`": `(1, 3, +2)`, `(2, 5, +1)`, `(4, 4, −3)`. În loc să modifici fiecare element, faci doar `d[x] += v; d[y + 1] -= v;`, iar la final calculezi sumele parțiale ale lui `d`. Ce vector obții? De ce funcționează? Cât costă `m` operații?

P23. Secvențe de sumă 0, fără să le încerci pe toate. Arată că `a[x..y]` are suma 0 exact atunci când `s[y] = s[x − 1]`. Folosește observația ca să numeri secvențele de sumă 0 din `(1, −1, 2, −2, 0)` doar din valorile `s[0..5]`. Ce structură de la fișa trecută te ajută?

P24. Adevărat sau fals? a) „Secvența de sumă maximă conține mereu elementul maxim al vectorului." **b)** „Dacă vectorul are măcar un element pozitiv, există o secvență de sumă maximă care începe cu un element pozitiv." Pentru afirmația falsă dă un contraexemplu, pentru cea adevărată o explicație.

P25. Mini-proiect: două capete. Toate valorile sunt **pozitive**. Găsește cea mai lungă secvență cu suma ≤ S, în O(n): capătul drept avansează la fiecare pas, iar cel stâng avansează cât timp suma depășește S. Test: `a = (2, 1, 3, 1, 1, 4)`, `S = 5` → `3`. De ce nu merge metoda dacă există și valori negative?

Probleme de lucru în clasă (pbinfo)

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

- [ ] #516 **sumsec1** — ușoară
  https://www.pbinfo.ro/probleme/516/sumsec1
- [ ] #523 **secvegale** — ușoară
  https://www.pbinfo.ro/probleme/523/secvegale
- [ ] #299 **sumesecv** — ușoară
  https://www.pbinfo.ro/probleme/299/sumesecv
- [ ] #297 **secvsummax** — medie
  https://www.pbinfo.ro/probleme/297/secvsummax

Temă pentru acasă (pbinfo)

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

- [ ] #518 **secvzero** — ușoară
  https://www.pbinfo.ro/probleme/518/secvzero
- [ ] #181 **secvcresc** — medie
  https://www.pbinfo.ro/probleme/181/secvcresc
- [ ] #2921 **sumesecv1** — medie
  https://www.pbinfo.ro/probleme/2921/sumesecv1
- [ ] #4478 **nr-max-de-zerouri** — ușoară
  https://www.pbinfo.ro/probleme/4478/nr-max-de-zerouri