FIȘĂ DE LUCRU · CLASA A X-A · FIȘA 4 · ACOPERĂ LECȚIA X.5

Căutare binară și interclasare

Durată: 90 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Urmărești valorile lui st, dr și mij și numeri pașii căutării binare, ca la grilele de BAC.
- Folosești „prima poziție ≥ x" și „ultima poziție ≤ x" pentru numărări și cel mai apropiat element.
- Interclasezi, intersectezi și reunești vectori sortați în O(n + m), inclusiv din fișier.

Ce trebuie să știi (o pagină)

Ce cauți (vector crescător `a[1..n]`)	Cod esențial	Dacă nu există
poziția lui `x`	`if (a[m] == x) poz = m; else if (a[m] < x) st = m + 1; else dr = m - 1;`	`poz = -1`
prima poziție cu `a[i] ≥ x`	`if (a[m] >= x) { rez = m; dr = m - 1; } else st = m + 1;`	`rez = n + 1`
ultima poziție cu `a[i] ≤ x`	`if (a[m] <= x) { rez = m; st = m + 1; } else dr = m - 1;`	`rez = 0`
câte elemente sunt `< x`	(prima poziție `≥ x`) − 1	
câte sunt în `[x, y]`	(prima `> y`) − (prima `≥ x`)	
STL	`lower_bound(a+1, a+n+1, x) - a` (prima `≥ x`), `upper_bound(...)` (prima `> x`)	dă `n + 1`
pași, cel mult	`n = 7 → 3`, `100 → 7`, `1000 → 10`, `10⁶ → 20`	

Combinare (doi vectori crescători)	La `a[i] == b[j]`	După bucla principală
interclasare	iei unul (oricare), avansezi doar în el	copiezi restul din ambii
intersecție	scrii o dată, `i++; j++;`	nimic
reuniune (fără duplicate)	scrii o dată, `i++; j++;`	copiezi restul
diferența `a − b`	sari peste, `i++; j++;`	copiezi restul lui `a`

ATENȚIE — GREȘELI FRECVENTE

Bucla este `while (st <= dr)`, iar actualizările sunt `m + 1` și `m - 1`. Cu `st < dr` pierzi elemente; cu `st = m` bucla se poate bloca. Și: căutarea binară pe un vector **nesortat** nu dă nicio eroare, doar răspunsuri greșite.

Nivelul 1 · Prezice

★ citește și prezice

P1. Căutarea binară clasică pe `a = (3, 8, 11, 15, 19, 24, 30, 36, 41)`. Completează tabelul pentru `x = 30`, apoi pentru `x = 4`. Ce valori au `st` și `dr` la final, când `x = 4`?

x	valorile lui `mij` (și `a[mij]`)	poz	pași
30			
4			

P2. „Prima poziție cu `a[i] ≥ x`" pe `a = (2, 4, 4, 4, 7, 9, 9, 12)`. Scrie valorile lui `mij` și rezultatul `rez` pentru `x = 4`, `x = 8`, `x = 13`. De ce, la `x = 4`, nu se oprește la prima valoare 4 întâlnită?

P3. Vectorul `(10, 20, 30, 40, 50, 60, 70)`. Pentru care valori din vector căutarea clasică se termină după **exact 1** pas? Dar după **3** pași? Câți pași face pentru o valoare care nu e în vector, de exemplu 45? Desenează „arborele" valorilor lui `mij`.

P4. Interclasează `a = (2, 6, 6, 11)` cu `b = (1, 6, 9)`. Sub fiecare element din rezultat scrie din ce vector vine (`a` sau `b`), întâi cu condiția `a[i] < b[j]`, apoi cu `a[i] <= b[j]`. Se schimbă vectorul rezultat? Dar șirul de litere?

P5. Ce afișează pentru `a = (1, 3, 4, 7, 9, 12)` și `b = (3, 5, 7, 8, 12, 15)`? De câte ori se execută corpul lui `while`?

```
int i = 1, j = 1;
while (i <= n && j <= m)
    if (a[i] == b[j]) { cout << a[i] << ' '; i++; j++; }
    else if (a[i] < b[j]) i++;
    else j++;
```

P6. Interclasare „de la coadă". Ce conține `c` pentru `a = (3, 8, 14)`, `b = (5, 6, 20, 25)`? În ce ordine au fost **scrise** elementele în `c`?

```
int i = n, j = m, k = n + m;
while (i >= 1 && j >= 1)
    if (a[i] > b[j]) c[k--] = a[i--];
    else c[k--] = b[j--];
while (i >= 1) c[k--] = a[i--];
while (j >= 1) c[k--] = b[j--];
```

P7. Pentru `a[1..7] = (1, 3, 5, 5, 5, 8, 10)`, cât valorează fiecare expresie? Ce înseamnă prima, în cuvinte?

```
upper_bound(a + 1, a + 8, 5) - lower_bound(a + 1, a + 8, 5)   // a)
lower_bound(a + 1, a + 8, 6) - a                               // b)
upper_bound(a + 1, a + 8, 10) - a                              // c)
lower_bound(a + 1, a + 8, 0) - a                               // d)
```

P8. Jocul „Ghicește numărul" între 1 și 100. Tu spui mereu `mij = (st + dr) / 2`, iar prietenul răspunde „mai mare / mai mic / da". Ce numere spui dacă numărul secret este 37? Dar 100? Dar 1? Care este numărul maxim de încercări pentru orice număr secret?

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Elevul a scris `while (st < dr && poz == -1)`, restul fiind codul clasic. Pe `a = (4, 8, 15, 16, 23, 42)`, ce afișează pentru `x = 42`? Dar pentru `x = 4`? Care dintre cele 6 valori din vector **nu** sunt găsite? Repară.

P10. Elevul a scris `else if (a[m] < x) st = m;` (restul e corect). Pe `a = (1, 3, 5, 7)`, pentru care valori `x` de la 0 la 8 programul **nu se mai oprește**? Explică pe cazul `x = 7`.

P11. Ce afișează pentru `a = (2, 5, 9, 12)` și `b = (3, 4, 20)`? Ce element lipsește și de ce? Adaugă ce trebuie.

```
int i = 1, j = 1, k = 0;
while (i <= n && j <= m)
    if (a[i] < b[j]) c[++k] = a[i++];
    else c[++k] = b[j++];
for (int t = 1; t <= k; t++) cout << c[t] << ' ';
```

P12. *Puzzle cu rânduri amestecate.* Reuniunea a două mulțimi sortate (fără duplicate în fiecare). Pune rândurile în ordine; un rând este în plus. Test: `(1, 4, 6, 9)` și `(2, 4, 9, 11, 13)` → `1 2 4 6 9 11 13`.

```
A)      else { c[++k] = a[i]; i++; j++; }
B)  while (j <= m) c[++k] = b[j++];
C)  while (i <= n && j <= m)
D)      if (a[i] < b[j]) c[++k] = a[i++];
E)  int i = 1, j = 1, k = 0;
F)      else { c[++k] = a[i]; c[++k] = b[j]; i++; j++; }
G)  while (i <= n) c[++k] = a[i++];
H)      else if (a[i] > b[j]) c[++k] = b[j++];
```

P13. Completează golurile ca `rez` să fie **ultima** poziție cu `a[i] ≤ x` (0 dacă nu există). Teste pe `(2, 4, 4, 7, 9)`: `x = 4 → 3`, `x = 1 → 0`, `x = 10 → 5`, `x = 8 → 4`.

```
int st = 1, dr = n, rez = ___;
while (st <= dr)
{
    int m = (st + dr) / 2;
    if (a[m] ______ x) { rez = m; _____________; }
    else _____________;
}
```

P14. Ce greșeală a făcut elevul? Anii în care s-a renovat școala sunt memorați **descrescător**: `(2026, 2019, 2011, 2004, 1998, 1990)`. Cu codul clasic, căutarea lui 2004 dă `-1`. Ce valori ia `mij`? Ce trebuie schimbat, fără să sortezi vectorul?

Nivelul 3 · Scrie cod

★★ scrie cod

La P15–P16 fiecare întrebare trebuie rezolvată în O(log n), iar la P17–P19 în O(n + m). Verifică **toate** testele.

P15. Pragul de admitere. Se dau `n` punctaje **crescătoare** și `q` praguri. Pentru fiecare prag `x`, câți candidați au cel puțin `x` puncte?

intrare	ieșire
`6` / `45 60 60 72 88 95` / `3` / `60 90 100`	`5 1 0`
`1` / `50` / `2` / `50 10`	`1 1`
`3` / `10 20 30` / `1` / `31`	`0`

P16. Cea mai apropiată benzinărie. Se dau kilometrii a `n` benzinării, crescător, și `q` poziții ale unor mașini. Pentru fiecare mașină, afișează kilometrul celei mai apropiate benzinării (la egalitate, cea din spate, cu kilometrul mai mic).

intrare	ieșire
`4` / `10 25 40 90` / `4` / `30 33 95 0`	`25 40 90 10`
`2` / `10 20` / `1` / `15`	`10`
`1` / `7` / `2` / `100 -5`	`7 7`

P17. Al k-lea din două clasamente. Se citesc `n m k`, apoi doi vectori crescători. Afișează al `k`-lea element din interclasarea lor, **fără** să construiești al treilea vector.

intrare	ieșire
`4 3 4` / `3 8 11 20` / `1 9 14`	`9`
`4 3 7` / `3 8 11 20` / `1 9 14`	`20`
`2 2 3` / `5 5` / `5 6`	`5`

P18. Prieteni comuni. Ana și Bogdan au listele de prieteni (ID-uri crescătoare, distincte). Afișează pe un rând câți prieteni comuni au, iar pe următorul ID-urile lor.

intrare	ieșire
`5 4` / `2 5 8 13 21` / `3 5 13 34`	`2` / `5 13`
`2 2` / `1 2` / `3 4`	`0`
`3 3` / `1 2 3` / `1 2 3`	`3` / `1 2 3`

P19. BAC, cu fișier. `bac.txt` conține `n m`, apoi `n` numere crescătoare și `m` numere crescătoare (n, m ≤ 10^6). Afișează pe ecran toate cele `n + m` numere, crescător. Memorează **doar** primul șir; pe al doilea îl prelucrezi pe măsură ce îl citești.

`bac.txt`	ecran
`3 4` / `1 5 9` / `2 5 6 12`	`1 2 5 5 6 9 12`
`1 2` / `10` / `1 2`	`1 2 10`
`2 1` / `3 4` / `100`	`3 4 100`

P20. Grădina pătrată. Ai `n` dale pătrate (n ≤ 10^{18}). Care este latura celei mai mari grădini pătrate pe care o poți pava? **Fără** `sqrt`: caută binar latura.

intrare	ieșire
`50`	`7`
`49`	`7`
`1`	`1`
`1000000000000000000`	`1000000000`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Detectivul invers. Pe vectorul `(5, 9, 14, 20, 26, 31, 40)`, căutarea clasică a luat valorile `mij = 4, 6, 5`, în această ordine. Ce valori naturale putea avea `x`? În care dintre ele s-a afișat o poziție?

P22. Muntele. Vectorul are valori distincte, întâi strict crescătoare, apoi strict descrescătoare (de exemplu `(1, 4, 9, 12, 7, 3)`). Găsește poziția vârfului în O(log n). Indiciu: ce îți spune comparația dintre `a[mij]` și `a[mij + 1]`? Verifică și pe vectori doar crescători sau doar descrescători.

P23. Merită să sortez? Ai `n = 10^5` numere nesortate și `q` întrebări „apare `x`?". Estimează numărul de operații pentru „caut secvențial de fiecare dată" și pentru „sortez o dată, apoi caut binar", când `q = 5` și când `q = 10^5`. Cam de la ce `q` merită să sortezi? Câți pași face căutarea binară pentru `n = 10^9`?

P24. Un vector întors. `a` e crescător, iar `b` e **descrescător**. Interclasează-i crescător fără să inversezi `b` și fără vector auxiliar pentru `b`. De unde pornește indicele din `b`? Test: `a = (2, 7, 10)`, `b = (12, 8, 7, 1)` → `1 2 7 7 8 10 12`.

P25. Mini-proiect: bușteni. Ai bușteni de lungimi `8, 5, 12` și vrei cel puțin `k = 5` bucăți de **aceeași** lungime întreagă `L`, cât mai mare (resturile se aruncă). Pentru un `L` dat, câte bucăți obții? De ce, dacă un `L` merge, merge și orice lungime mai mică? Folosește observația ca să cauți binar cel mai mare `L`. Ce iese pentru `k = 30`?

Probleme de lucru în clasă (pbinfo)

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

- [ ] #4856 **cb-0** — ușoară
  https://www.pbinfo.ro/probleme/4856/cb-0
- [ ] #508 **cautare-binara** — medie
  https://www.pbinfo.ro/probleme/508/cautare-binara
- [ ] #241 **interclasare** — ușoară
  https://www.pbinfo.ro/probleme/241/interclasare

Temă pentru acasă (pbinfo)

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

- [ ] #507 **cautare** — ușoară
  https://www.pbinfo.ro/probleme/507/cautare
- [ ] #2276 **cb** — ușoară
  https://www.pbinfo.ro/probleme/2276/cb
- [ ] #250 **interclasare1** — medie
  https://www.pbinfo.ro/probleme/250/interclasare1
- [ ] #530 **multimi1** — medie
  https://www.pbinfo.ro/probleme/530/multimi1