FIȘĂ DE LUCRU · CLASA A XI-A · FIȘA 3 · ACOPERĂ LECȚIA XI.5

Divide et impera: căutare binară, MergeSort, QuickSort, Hanoi

Durată: 90 min Nivel: mate-info

CE VEI ȘTI DUPĂ LECȚIE

- Desenezi arborele de apeluri pe intervale [st, dr] și spui ce întoarce, ce afișează și de câte ori se apelează o funcție D&I.
- Urmărești de mână căutarea binară, interclasările din MergeSort, partiționarea din QuickSort și mutările de la Hanoi.
- Scrii funcții D&I în care pasul de combinare este partea care contează.

Ce trebuie să știi (o pagină)

Ce	Cod / regulă	Rezultat de reținut
schema	`if (st == dr) return …; mij = (st + dr) / 2;` stânga `[st, mij]`, dreapta `[mij + 1, dr]`, combini	intervalele **nu** se suprapun
număr de apeluri, până la un element	arbore cu n frunze	**2n − 1** apeluri
căutare binară (vector sortat)	doar **o** jumătate: `[st, mij − 1]` sau `[mij + 1, dr]`	cel mult ~$\log_2$ n + 2 apeluri
MergeSort	sortezi stânga, sortezi dreapta, **interclasezi** în `aux`, copiezi înapoi	O(n log n), stabil (`<=`)
QuickSort	pivot `a[(st + dr) / 2]`; `while (a[i] < pivot) i++; while (a[j] > pivot) j--;` schimbi dacă `i <= j`	apoi `[st, j]` și `[i, dr]`
Hanoi	`hanoi(n−1, a, c, b); a→c; hanoi(n−1, b, a, c);`	**2^n − 1** mutări
unde afișezi	`cout` înainte de apeluri / între ele / după ele	ordine diferită a nodurilor

SFAT

La orice grilă D&I desenează arborele cu intervalele `[st, dr]` în noduri. Valorile se completează **de jos în sus** (de la frunze), iar afișările se citesc în ordinea în care parcurgi arborele.

Nivelul 1 · Prezice

★ citește și prezice

P1. Pentru `v = (5, 2, 9, 4, 7)` (indexat de la 1), cât este `f(1, 5)` și câte apeluri se fac?

```
int f(int st, int dr) {
    if (st == dr) return v[st];
    int m = (st + dr) / 2;
    int a = f(st, m), b = f(m + 1, dr);
    return a > b ? a - b : b - a;
}
```

P2. Căutarea binară din lecție, pe `a = (1, 3, 4, 7, 9, 11, 15, 20)`. Completează tabelul pentru `x = 15` și pentru `x = 10` (rânduri câte sunt necesare). Ultimul apel pentru `x = 10` are `st > dr`.

apel	st	dr	mij	a[mij]	decizie

P3. MergeSort pe `(8, 3, 5, 1, 7, 2, 6, 4)`. Scrie **vectorul întreg** după fiecare interclasare, în ordinea în care se fac. Câte interclasări sunt?

P4. Prima partiționare QuickSort pe `(6, 1, 8, 4, 9, 3, 7)`: care este pivotul, ce perechi se schimbă, cum arată vectorul și cât sunt `i` și `j` la final? Ce apeluri urmează?

P5. La `hanoi(4, 'A', 'B', 'C')`: câte mutări sunt în total? Care este a 5-a mutare? Dar a 8-a? De câte ori se mută discul cel mai mic?

P6. Ce afișează `f(1, 7)`? Ce afișează dacă muți `cout` **între** cele două apeluri?

```
void f(int st, int dr) {
    if (st > dr) return;
    int m = (st + dr) / 2;
    cout << m << ' ';
    f(st, m - 1);
    f(m + 1, dr);
}
```

P7. Pentru `a = (3, 5, 2, 8, 6, 7, 1)`, cât este `g(1, 7)`? Ce numără funcția? De ce este nevoie de termenul `(a[m] > a[m + 1])`?

```
int g(int st, int dr) {
    if (st == dr) return 0;
    int m = (st + dr) / 2;
    return g(st, m) + g(m + 1, dr) + (a[m] > a[m + 1]);
}
```

P8. Completează tabelul (pentru căutarea binară: numărul **maxim** de apeluri al funcției din lecție, când `x` nu este în vector).

n	1	2	5	8	10	1000
apeluri `suma(1, n)` D&I						
apeluri căutare binară (max)						
mutări Hanoi cu n discuri						—

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Elevul a împărțit intervalul așa: `return suma(st, m) + suma(m, dr);`. Pentru un vector cu 2 elemente programul se oprește cu eroare. Scrie primele apeluri și repară.

P10. În căutarea binară un elev a scris `if (x < a[mij]) return caut(st, mij, x);`. Pe `a = (2, 4, 6)`, pentru `x = 7` răspunde corect `false`, dar pentru `x = 1` nu se mai oprește. Explică de ce, scriind apelurile.

P11. În MergeSort, ultima linie a fost scrisă `for (t = 1; t <= k; t++) a[t] = aux[t];`. Pe `(5, 2, 9, 1, 7, 3)` programul afișează `1 1 3 7 7 3`. Ce greșeală este și cum o repari?

P12. *Puzzle cu rânduri amestecate.* Funcția verifică D&I dacă `a[st..dr]` este crescător. Pune rândurile în ordine; un rând este în plus. Dă un vector de 4 elemente pe care rândul în plus ar da răspuns greșit.

```
A)      int m = (st + dr) / 2;
B)  }
C)      return sortat(st, m) && sortat(m + 1, dr);
D)  bool sortat(int st, int dr) {
E)      return sortat(st, m) && sortat(m + 1, dr) && a[m] <= a[m + 1];
F)      if (st == dr) return true;
```

P13. Completează golurile: funcția întoarce **prima poziție** `p` din `[st, dr + 1]` cu `a[p] >= x` (vector sortat). Ce întoarce pentru `a = (2, 4, 4, 7, 9)` și `x = 4`, `x = 5`, `x = 10`, `x = 1`?

```
int prima(int st, int dr, int x) {
    if (st > dr) return ___;
    int m = (st + dr) / 2;
    if (a[m] >= x) return prima(___, ___, x);
    return prima(___, ___, x);
}
```

P14. Un elev a scris Hanoi cu primul apel `hanoi(n - 1, a, b, c);` (fără să schimbe rolurile tijelor). Ce afișează pentru `n = 2`? De ce mutările nu sunt permise?

Nivelul 3 · Scrie cod

★★ scrie cod

Fiecare funcție cerută este **divide et impera** (sau căutare binară recursivă). Verifică toate testele.

P15. Cea mai friguroasă zi. Se dau `n` temperaturi. Afișează temperatura minimă și **prima** zi în care a fost atinsă, cu o funcție D&I care întoarce **poziția** minimului. Teste: `6` / `3 -2 5 -7 0 -7` → `-7 4`; `1` / `12` → `12 1`; `4` / `5 5 5 5` → `5 1`.

P16. Ghicește numărul. Calculatorul ghicește un număr secret din `[1, 100]`: spune mijlocul intervalului, află „mai mic / mai mare” și continuă în jumătatea bună. Afișează toate încercările. Teste: `37 → 50 25 37`; `100 → 50 75 88 94 97 99 100`; `1 → 50 25 12 6 3 1`.

P17. Mutarea a k-a. Se dau `n ≤ 60` și `k`. Afișează a `k`-a mutare din `hanoi(n, 'A', 'B', 'C')` **fără** să le generezi pe toate (indiciu: mutarea discului mare are numărul 2^{n-1}). Teste: `3 4 → A->C`; `4 5 → C->A`; `10 1000 → B->A`.

P18. Clasamentul. Se dau `n` sportivi (nume și punctaj). Afișează numele în ordinea **descrescătoare** a punctajelor, cu MergeSort; la egalitate rămâne ordinea din intrare.

intrare	ieșire
`5` `Ana 90` `Bogdan 75` `Cris 90` `Dan 100` `Eva 75`	`Dan Ana Cris Bogdan Eva`
`3` `X 1` `Y 1` `Z 1`	`X Y Z`

P19. Latura terenului. Pentru `n ≤ 10¹²`, afișează cel mai mare `k` cu `k · k ≤ n`, prin căutare binară recursivă (fără `sqrt`). Teste: `10 → 3`; `16 → 4`; `0 → 0`; `999999999999 → 999999`; `1000000000000 → 1000000`.

P20. Vârful muntelui. Altitudinile de pe un traseu cresc strict, apoi scad strict (oricare parte poate lipsi). Găsește poziția vârfului în O(log n). Teste: `7` / `1 3 8 12 9 4 2` → `4`; `3` / `5 4 3` → `1`; `3` / `1 2 3` → `3`.

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Problema inversă. Pentru funcția din P1, găsește un vector de 4 cifre **distincte** din `{1, …, 9}` pentru care `f(1, 4) = 0`. Care este cea mai mare valoare posibilă a lui `f(1, 4)` pentru astfel de vectori? Justifică fără să le încerci pe toate.

P22. Hanoi cu tije vecine. Acum discurile se pot muta **doar între tije vecine** (A↔B și B↔C, niciodată direct A↔C). Gândește recursiv cum muți `n` discuri de pe A pe C și scrie relația pentru numărul de mutări `M(n)`. Calculează `M(1)`, `M(2)`, `M(3)` și ghicește formula.

P23. Maxim și minim cu mai puține comparații. Naiv, maximul și minimul a `n` numere costă `2(n − 1)` comparații. Scrie o funcție D&I care le întoarce pe amândouă (prin referință), tratând separat intervalele de 2 elemente (o singură comparație). Câte comparații face pentru `n = 8`? Dar pentru `n = 16`?

P24. De ce trebuie sortat? Pe vectorul **nesortat** `(5, 1, 9, 3, 7)` aplici căutarea binară din lecție pentru fiecare dintre cele 5 valori ale lui. Pe care le găsește? Explică de ce exact acestea.

Probleme de lucru în clasă (pbinfo)

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

- [ ] `#1015` **sumvec** — ușoară
  https://www.pbinfo.ro/probleme/1015/sumvec
- [ ] `#1019` **maxim6** — ușoară
  https://www.pbinfo.ro/probleme/1019/maxim6
- [ ] `#1155` **cautarebinara** — medie
  https://www.pbinfo.ro/probleme/1155/cautarebinara

- [ ] #2527 **hanoi** — ușoară
  https://www.pbinfo.ro/probleme/2527/hanoi

Temă pentru acasă (pbinfo)

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

- [ ] #1018 **cntimpare** — ușoară
  https://www.pbinfo.ro/probleme/1018/cntimpare
- [ ] #1023 **cmmdc3** — ușoară
  https://www.pbinfo.ro/probleme/1023/cmmdc3
- [ ] #1152 **verificareordonatdivimp** — ușoară
  https://www.pbinfo.ro/probleme/1152/verificareordonatdivimp
- [ ] #1025 **mergesort** — medie
  https://www.pbinfo.ro/probleme/1025/mergesort
- [ ] #3206 **nrinversiuni** — medie
  https://www.pbinfo.ro/probleme/3206/nrinversiuni