

Seminar 2

Problemă rezolvată de un algoritm.

Complexitatea algoritmilor.

Ștefan Ciobâcă, Dorel Lucanu
Universitatea “Alexandru Ioan Cuza”, Iași

March 4, 2022

1. Implementați în ALK algoritmul lui Euclid prin scăderi repetate, împărțiri repetate și căutarea explicită a celui mai mare divizor comun (vedeți fișa de exerciții a primului seminar). Calculați timpul de execuție în modelul uniform și în modelul logaritmnic pentru algoritmi de mai sus pentru datele de intrare $a = 1$ și $b = n$ (unde n este un număr natural în funcție de care trebuie calculat timpul de calcul). Puteți folosi metoda prezentată în cadrul cursului, slide-urile *Computing cost functions with Alk interpreter*.
2. Enunțați problema sortării în ordine crescătoare a unui șir de numere naturale.
 - (a) Scrieți în Alk algoritmi bubble-sort, insertion-sort și selection sort pentru problema de mai sus.
 - (b) Arătați că algoritmi de mai sus rezolvă într-adevăr problema.
 - (c) Calculați timpul de execuție în cele două modele de calcul pentru algoritmi de sortare de mai sus pentru intrarea dată de vectorul $1, 2, \dots, n$ și pentru $n, n-1, \dots, 1$, unde n este un parametru în funcție de care trebuie calculat timpul de calcul.
3. Operații pe numere mari.
 - (a) Enunțați problemele adunării/înmulțirii a două numere naturale, numere reprezentate printr-un șir de cifre.
 - (b) Scrieți algoritmi pentru problemele de mai sus.
 - (c) Calculați timpul de calcul în cazul cel mai nefavorabil pentru cei doi algoritmi de mai sus, în funcție de dimensiunea datelor de intrare.
 - (d) Problemă mai grea: scrieți algoritmul lui Karatsuba pentru înmulțirea a două numere naturale, de complexitate $O(n^{\log_2(3)})$.

4. Găsiți soluții cât mai eficiente din punct de vedere al timpului de calcul și al necesarului de memorie pentru următoare problemă: dându-se un număr natural n și $n - 1$ numere distincte între 1 și n să se determine numărul lipsă. Formalizați problema ca pereche **Input-Output**. Arătați că algoritmul (sau algoritmii) propuși rezolvă într-adevăr problema; calculați timpul de calcul și necesarul de memorie.
5. Același exercițiu ca mai sus, dar de data aceasta un număr apare de două ori și un număr lipsește. Formalizați problema.
6. Să se construiască arbori de decizie pentru sortare pentru fiecare dintre următorii algoritmi: heapsort, mergesort, bubblesort, pentru $n = 3$.
7. Să se dea un exemplu de arbore de decizie (cu noduri $i?.j$) care **nu rezolvă** problema sortării.
8. Formalizați următoarele reduceri:
 - (a) cmmmc la cmmdc,
 - (b) egalitatea de mulțimi la sortare,
 - (c) incluziunea de mulțimi la sortare,
 - (d) disjuncția de mulțimi la sortare.
9. Să se găsească limite inferioare pentru problemele:
 - (a) egalitatea de mulțimi,
 - (b) incluziunea de mulțimi,
 - (c) disjuncția de mulțimi,
 - (d) interclasarea a doua secvențe ordonate.
10. Cum definim complexitatea $O(f(n))$ a unei probleme? Dar $\Omega(f(n))$?
11. Ce înseamnă algoritm optim pentru o problemă?
12. Arătați că dacă un algoritm A rezolvă o problemă P în timp $O(f)$ și $f = O(g)$, atunci problema P are complexitatea $O(g)$.
13. Justificați, plecând de la definiție, că $0.5n^2 - 3n = \theta(n^2)$.
14. Arătați că dacă f este polinom de grad k , atunci $f(n) \in \theta(n^k)$.
15. Arătați că $\theta(f) = O(f) \cap \Omega(f)$ pentru orice funcție f .
16. Fie $T(n) = \begin{cases} \theta(1) & \text{dacă } n = 1 \\ 2T(n/2) + \theta(n) & \text{dacă } n > 1. \end{cases}$
 Arătați că $T(n) = \theta(n \log(n))$.

17. Arătați prin inducție că următoarea funcție întoarce rezultatul $3^n - 2^n$:

```
f(n) {  
    if (n == 0 || n == 1) {  
        return n;  
    }  
    return 5 * f(n - 1) - 6 * f(n - 2);  
}
```

18. Scrieți în Alk algoritmul căutării Fibonacci pentru problema căutării unei valori într-un vector sortat și testați-l pe câteva exemple. (google search). Analizați complexitatea algoritmului relativ la complexitatea căutării binare obișnuite. Sunt algoritmi la fel de eficienți din punct de vedere asimptotic în cazul cel mai nefavorabil? Ce concluzie se poate trage despre complexitatea problemei căutării într-un vector ordonat?
19. Implementați în Alk algoritmul mergesort și scrieți un număr de teste relevante care să arate că algoritmul funcționează conform descrierii problemei (cel puțin pentru cazurile de test).