Seminar 8
Căutarea peste șiruri.

Ștefan Ciobâcă, Dorel Lucanu
Universitatea "Alexandru Ioan Cuza", Iași

Săptămâna 11 Aprilie - 15 Aprilie 2016

1. Implementați în Alk algoritmul naiv pentru căutarea peste șiruri.
2. Folosiți editorul favorit pentru a crea un fișier text cu conținutul "abab...ab" de dimensiune cât mai mare (cât de mare suportă editorul – orientativ 10-100 MB). Inserați un "a" undeva în mijlocul fișierului. Folosiți funcția de căutare pentru a căuta, pe rând, șirurile "aa", "baa", "abaa", "babaa", "ababaa", etc. Cât de rapidă este funcția de căutare a editorului în funcție de dimensiunea pattern-ului?
3. Implementați în Alk algoritmul Rabin-Karp.
4. Implementați în Alk algoritmul Boyer-Moore (cu regula caracterului rău).
5. Care este cazul cel mai nefavorabil pentru algoritmul naiv?
6. Care este cazul cel mai favorabil pentru algoritmul naiv?
7. Algoritmul naiv este corect și dacă înlocuim $i < n$ cu $i < n - m$ în bucla `for`? Complexitatea algoritmului se schimbă în acest caz?
8. Calculați hash-urile pattern-ului p, precum și a tuturor subșirurilor de lungime m ale lui s pentru următoarele valori: $p = aba$, $s = aabbababbabab$, $q = 3$. Aceeași cerință pentru $q = 7$.
9. Încercați să găsiți s, p și q astfel încât timpul de rulare al algoritmului Rabin-Karp să fie $O(n \cdot m)$.
10. Fie $q = 101$. Încercați să găsiți s și p astfel încât timpul de rulare al algoritmului Rabin-Karp pe s și p să fie $O(n \cdot m)$.
11. Enunțați problema căutării unei matrici într-o altă matrice.
12. Scrieți algoritmul naiv de căutare a unei matrici într-o altă matrice. Ce complexitate are algoritmul?
13. Adaptați algoritmul Rabin-Karp pentru a rezolva problema căutării unei matrici într-o altă matrice.
    (a) Găsiți o funcție hash corespunzătoare.
    (b) Explicați cum se calculează funcția hash când poziția matricii căutate se modifică cu o unitate.
    (c) Ce complexitate are algoritmul găsit?

14. Se consideră următoarea problemă:

    Input: două șiruri s și t, amândouă de lungime n *Output*: "da", dacă s este o *rotație* a lui t și "nu" altfel

    De exemplu, pentru intrarea $s = ABBA$ și $t = BAAB$, răspunsul corect este "da". Pentru intrarea $s = ABA$ și $t = BAB$, răspunsul este "nu".

    Reduceți problema la problema căutării. Concluzionați că această problemă se poate rezolva în timp liniar.

15. Două șiruri s și t sunt anagrame dacă orice caracter din alfabet apare de același număr de ori atât în s cât și în t. De exemplu $s = aba$ și $t = aab$ sunt anagrame, dar $s = abba$ și $t = aba$ nu sunt anagrame.

    Găsiți un algoritm cât mai eficient care caută toate anagramele unui șir p într-un alt șir s.

    De exemplu, dacă $p = aba$ și $s = baab$, anagrame a lui p apar la pozițiile 0 și 1.

    Hint: încercați să modificați algoritmul Rabin-Karp.

16. Dați un exemplu de date de intrare pentru care algoritmul Boyer-Moore se execută în timp mai mare decât liniar.