Lista zadań. Nr 9. 7 czerwca 2009

Algorytmy i Struktury Danych

IIUWr. II rok informatyki.

1. (2pkt) Rozważmy ciało F_2 liczb całkowitych modulo 2. Ułóż algorytm, który mnoży macierze $n \times n$ nad F_2 w czasie $O(n^{2.81}/(\log n)^{0.4})$.

2. (1pkt) Opisz, jak skonstruować automat, który dla dwóch danych wzorców P i P' znajduje wszystkie ich wystąpienia w tekście. Spróbuj zminimalizować liczbę stanów w Twoim automacie.

3. (2pkt) Słowa Fibonacciego definiujemy w następujący sposób:
$$F_0 = a \quad F_1 = b \quad \text{oraz } F_{k+1} = F_k F_{k-1} \text{ dla } k \geq 1.$$
Skonstruuj algorytm, który sprawdza w czasie liniowym, czy dany tekst jest słowem Fibonacciego. Postaraj się, by Twój algorytm używał możliwie najmniejszej pamięci pomocniczej.

4. (2pkt)Załóżmy, że wzorzec P może zawierać znak $\diamondsuit$ (tzw. gap character). Znak ten jest zgodny z dowolnym podsłowem (także z podsłowem pustym). Na przykład, wzorzec $ab\diamondsuit ba\diamondsuit c$ występuje w słowie $cabccbacbacab$ jako
$$c\,\underbrace{ab}_{ab}\underbrace{cc}_{\diamondsuit}\underbrace{ba}_{ba}\underbrace{cba}_{\diamondsuit}\underbrace{c}_{c}\,ab$$
a także jako
$$c\,\underbrace{ab}_{ab}\underbrace{ccbac}_{\diamondsuit}\underbrace{ba}_{ba}\underbrace{\quad}_{\diamondsuit}\underbrace{c}_{c}\,ab.$$
Podaj algorytm znajdujący wystąpienie takiego wzorca w danym tekście T (oczywiście zakładamy, że $\diamondsuit$ nie występuje w T).

5. (1pkt) Podaj algorytm, który w czasie liniowym określa, czy tekst T powstał przez przesunięcie cykliczne tekstu T'.

Zadania dodatkowe

1. (2pkt) Przedstaw algorytm Schoenhage'go-Strassena mnożenia długich liczb.

2. (3pkt) Pokaż, że problem sprawdzania, czy dane słowo należy do języka generowanego przez gramatykę bezkontekstową w normalnej postaci Chomsky'ego jest nie trudniejszy od problemu mnożena macierzy . Dodatkowe 3 pkt możesz uzyskać jeśli pokażesz odwrotną zależność.

Zadania dodatkowe - nie będą rozwiązywane na ćwiczeniach

1. Niech $\Sigma = \{a, b\})$.
   - Skonstruuj automat skończony dla wzorca $P = aabab$ i prześledź jego działanie na tekście $T = aaababaabaababaab$.
   - Skonstruuj automat skończony dla wzorca $ababbabbababbababbabb$.

2. Oblicz funkcję prefiksową π dla wzorca $ababbabbababbababbabb$ (alfabetem jest $\Sigma = \{a, b\}$).

3. Mówimy, że wzorzec P *nienakładalny* jeśli $P_k \sqsupset P_q$ implikuje $k = 0$ lub $k = q$. Opisz jak wygląda automat skończony dla takiego wzorca.

4. Opracuj efektywny algorytm wyszukiwania wzorca dla przypadku, gdy długość wzorca nie przekracza długości słowa maszynowego, który oparty jest na idei wyliczania wektora bitowego $S[1..m]$, takiego, że

$$S[i] \text{ wtedy i tylko wtedy, gdy } P[1..i] = T[j-i+1..j]$$

po wczytaniu j-tej litery tekstu. Ponieważ S mieści się w słowie maszynowym, użyj do jego obliczania długich operacji logicznych.

5. Opracuj efektywną metodę wyliczania przesunięcia jakie powinien zrobić algorytm Boyera-Moore'a stosując heurystykę "dobry suffix".

6. Podaj efektywny algorytm obliczający funkcję przejścia δ dla automatu M_P. Twój algorytm ma działać w czasie $O(m|\Sigma|)$.
Wskazówka: Udowodnij, że $\delta(q,a) = \delta(\pi(q),a)$, jeśli $q = m$ lub $P[q+1] \neq a$.

7. Podaj algorytm, który dla danego słowa znajduje najdłuższe powtarzające się w nim podsłowo.

8. Podaj algorytm wyszukiwania wszystkich wystąpień wzorców P_1 i P_2 w danym tekście T.

9. Algorytm Karpa-Rabina łatwo uogólnia się na przypadek, gdy tekst i wzorzec są dwuwymiarowe. Podaj szczegóły niezbędnych modyfikacji.

10. Oblicz dyskretną transformację Fouriera w pierścieniu liczb zespolonych wektorów:
    (a) [0,1,2,3,4],
    (b) [1,2,0,2,0,0,0,1]

11. Załóżmy, że zamiast wykonywać n-elementową FFT nad ciałem liczb zespolonych (gdzie n jest parzyste), używamy pierścienia Z_m liczb całkowitych modulo m, gdzie $m = 2^{\frac{tn}{2}} + 1$ i t jest dowolną liczbą całkowitą. Jako n-tego pierwotnego pierwotnego pierwiastka z jedności używamy, zamiast ω_n, $w = 2^t$ modulo m. Pokaż, że DFT i odwrotna DFT są w tym systemie dobrze zdefiniowane.