Wyszukiwanie informacji w sieci, Lista 3 **19.03.2007**

(zadania na ćw. w dniach 23.03 i 26.03.2007)

1. [1] Kody γ są stosunkowo mało wydajne dla dużych liczb, z uwagi na unarne kodowanie długości binarnej reprezentacji liczby. Kody δ (które oznaczać też będziemy jako γ_2) różnią się od kodów γ tym, że długość binarnej reprezentacji liczby kodowana jest przy użyciu kodów γ.

   (a) Dla jakich liczb kody δ są krótsze od kodów γ?

   (b) Podaj kolejne uogólnienia kodów γ, analogicznie do kodu δ, tworząc rodzinę kodów $\gamma_2, \gamma_3, \ldots$ Dla $i = 3, 4, \ldots, 10$ wyznacz minimalną wartość liczby k_i taką że kod γ_i dla k_i jest krótszy od kodu γ_{i-1} dla k_i.

2. [1] Niech p_i będzie prawdopodobieństwem pojawienia się w danych liczby i dla $i \in \mathcal{N}$. Niech K będzie kodem prefiksowym, w którym długość kodu liczby i równa jest k_i. Średnią długością kodu K nazywamy wyrażenie

$$\sum_i p_i k_i.$$

   Kod jest *optymalny* dla prawdopodobieństw $p_1, p_2, \ldots$, gdy jego średnia długość jest najmniejsza wśród wszystkich kodów prefiksowych.

   Z teorii informacji wiadomo, że średnia długość każdego kodu prefiksowego jest nie mniejsza niż

$$\sum_i p_i \log_2 \left(\frac{1}{p_i}\right).$$

   Czy istnieją takie prawdopodobieństwa $p_1, p_2, \ldots$ dla których kod unarny jest optymalny? A kod γ?

3. [0.5] W kalkulacjach dotyczących rozmiaru skompresowanego indeksu zakładaliśmy, że term występujący k razy w kolekcji złożonej z n dokumentów, występuje w równych odstępach n/k. Załóżmy teraz, że odstępy są nierówne. Zwiększy to czy zmniejszy rozmiar indeksu przy zastosowaniu kodowania γ?

4. [1] Przy uproszczonych założeniach dotyczących częstości występowania termów i ich rozmieszczenia w dokuemtach, rozmiar indeksu (list adresów) oszacowaliśmy jako

$$\sum_{j=1}^{\frac{M}{Lc}} \frac{2Nc \log_2 j}{j},$$

   gdzie

   - N to liczba dokumentów w korpusie,
   - M to liczba termów w korpusie,
   - c to stała wyznaczona na podstawie N, M i prawa Zipf'a,

- L to długość dokumentu, czyli liczba termów w dokumencie (zakładamy, że wszystkie dokumenty mają tę samą długość).

W oparciu o te same założenia, oszacuj rozmiar *pozycyjnego* indeksu odwróconego.

5. [2] Rozważmy następujący sposób reprezentacji słownika. Szukamy najkrótszego ciągu znaków t takiego, że każdy element słownika jest podsłowem t (jego wystąpienie w t jest niekoniecznie rozłączne z wystąpieniami innym elementów słownika). Natomiast w tablicy termów, dla każdego termu przechowujemy jego długość i pozycję jego wystąpienia w t.

   Problem wyznaczenia najkrótszego słowa t, którego podsłowami są podane słowa $w_1, \ldots, w_n$ nazywany jest problemem superstringu. Jest on NP-zupełny.

   Podaj wielomianowy algorytm, który dla danych stringów $w_1, \ldots, w_n$ znajduje superstring słów $w_1, \ldots, w_n$, który jest co najwyżej $O(\log n)$ razy dłuższy od najkrótszego superstringu.

   Wskazówka. Algorytm zachłanny, redukcja do problemu *set cover*. (p. V. Vazirani, *Approximation algorithms*, Springer).

6. [1] Omawiany na wykładzie kod Golomba wygląda w rzeczywistości nieco inaczej. Wykorzystuje się w nim fakt, że dla liczb z zakresu $[2^n, 2^{n+1}-1]$ można znaleźć kod prefiksowy, w którym zamiast kodować wszystkie liczby binarnie na $n+1$ bitach, część liczb kodujemy na n bitach a pozostałe na $n+1$ bitach.

   Przedstaw jak dokładnie wygląda kodowanie Golomba, podaj dla jakich wartości para metru m nie różni się ono od wersji przedstawionej na wykładzie. Ponadto, pokaż jak dla konkretnego kodu Golomba wyznaczyć prawdopodobieństwa $p_1, p_2, \ldots$, przy których ten kod będzie optymalny (wystarczy, że podasz jeden "zestaw" prawdopodobieństw dla danego kodu). Uzasadnij optymalność dla podanych przez Ciebie prawdopodieństw.