Repetytorium: lista zagadnień nr 3 *18 października 2005*

Algorytmy i Struktury Danych

METODA REDUKCJI

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. Przypomnij sobie algorytm Hoare'a rozwiązujący problem wyboru (znajdowanie k-tego co do wielkości elementu w nieuporządkowanym zbiorze liczb pamiętanym w nieuporządkowanej tablicy $T[0 \ldots n-1]$, gdzie $0 \leq k < n$). Przeanalizuj czas działania tego algorytmu w najlepszym przypadku, w najgorszym przypadku i w przypadku średnim.
   Uwaga: Przyjmij, że wszystkie elementy zbioru są różne. Wobec tego procedura podziału `partition3` dzieli zbiór wejściowy T na trzy podzbiory T_1, T_2 i T_3, przy czym $|T_2| = 1$.
   Uwaga: Przyjmij, że element dzielący jest wybierany z rozpatrywanego zbioru z jednakowym prawdopodobieństwem (rozkład jednostajny) równym $\frac{1}{n}$, gdy $|T| = n$. W analizie przypadku średniego możesz więc założyć, że po podziale $|T_1|$ (podobnie jak $|T_3|$) jest z prawdopodobieństwem $\frac{1}{n}$ równa $0, 1, \ldots, n-1$.

2. Przypomnij sobie algorytm wyszukiwania binarnego. Rozwiązywał on problem sprawdzania, czy w n-elementowej posortowanej tablicy T znajduje się żądana wartość x. Zmodyfikuj ten algortm w taki sposób, aby podawał on pozycję w tablicy (liczba z zakresu $0 \ldots n-1$), na której wartość ta jest zapisana, albo liczbę -1, gdy ratość ta nie występuje w tablicy. Czy twój algorytm bardzo się uprości, jeśli będziemy wiedzieli, że wartość x zawsze znajduje się w tablicy T?

3. Udowodnij, że dolna granica czasowa dla problemu wyszuliwania w zbiorze uporządkowanym, czyli sprawdzania, czy w n-elementowej posortowanej tablicy T znajduje się żądana wartość x), wynosi $\Omega(\log n)$. Za operację podstawową przyjmij porównywanie elementów tablicy.

4. Dane są dwie n elementowe tablice liczb R i S, obydwie uporządkowane niemalejąco. Podaj algorytm, który wyznaczy taki parametr podziału k tych tablic, który to podział będzie się charakteryzował następującymi własnościami:

   (a) Tablica R będzie podzielona na podtablice $R_0[0 \ldots k-1]$ i $R_1[k \ldots n-1]$.
   (b) Tablica S będzie podzielona na podtablice $S_0[0 \ldots n-k-1]$ i $S_1[n-k \ldots n-1]$.
   (c) $(x_0 \in R_0 \cup S_0) \wedge (x_1 \in R_1 \cup S_1) \Rightarrow x_0 \leq x_1$

   Złożoność czasowa twojego rozwiązania powinna być rzędu $O(\log n)$.