Algorytmy i struktury danych

Lista zadań 3

Zadanie 1

Ile (dokładnie) porównań wykona algorytm `insertion_sort` w wersji z wartownikiem (liczbą zapisaną pod adresem `t[-1]`), jeśli dane $(a_1, \ldots, a_n)$ o rozmiarze n zawierają k inwersji. Liczba inwersji to liczba takich par (i, j), że $i < j$ i $a_i > a_j$. Jaka jest maksymalna możliwa liczba inwersji dla danych rozmiaru n? Wylicz "średnią" złożoność algorytmu, jaka średnią z maksymalnej i minimalnej ilości porównań jaką wykona.
Uwaga: Prawdziwą średnią złożoność oblicza się, jako średnią po wszystkich możliwych permutacjach danych wejściowych.

$$\text{Dokładna ilość porównań: } n-1+k$$
$$\text{Minimalna ilość porównań: } n-1+0 = n-1$$
$$\text{Maksymalna ilość porównań: } n-1+\frac{n(n-1)}{2} = \frac{n^2+n-2}{2}$$
$$\text{Średnia ilość porównań: } \frac{n-1+\frac{n^2+n-2}{2}}{2} = \frac{n^2+3n-4}{4}$$

Zadanie 3

(a) Ile co najwyżej porównań wykona procedura `insertion_sort` działająca na ostatnim etapie `bucket_sort` zakładając, że `bucket_sort` korzysta z `k` pomocniczych kolejek, i że do każdej z nich wpadła taka sama ilość elementów? Zakładamy wersję z wartownikiem na pozycji `t[-1]`.

$$n-1+k\frac{\frac{n}{k}(\frac{n}{k}-1)}{2}$$

(b) Podaj uproszczony wynik dla $k = n/2$, $k = n/4$, $k = n/10$ oraz $k = \sqrt{n}$. Następnie każdy z tych wyników zapisz też w notacji asymptotycznej $O(f(n))$.

$$3n/2-1 = O(n) \qquad 5n/2-1 = O(n)$$
$$11n/2-1 = O(n) \qquad \frac{n^{3/2}+n-2}{2} = O(n^{3/2})$$

(c) Jaki będzie wynik, gdy wszystkie klucze wpadną do tego samego kubełka?

$$n-1+\frac{n(n-1)}{2}$$

Zadanie 4

Iteracyjna wersja procedury mergesort polega na scalaniu posortowanych list. Zaczyna łączenia pojedynczych elementów w posortowane pary, potem par w czwórki itd. aż do połączenia dwóch ostatnich list w jedną.

(a) Zakładając, że rozmiar tablicy jest potęgą dwójki $n = 2^k$ oraz, że procedura merge wykonuje dokładnie tyle porównań, ile jest elementów po scaleniu, oblicz ile dokładnie porównań wykona cały algorytm.

$$\sum_{i=1}^{k} 2^k = k2^k = kn$$

(b) Ile razy jest wywołana procedura merge w trakcie działania całego algorytmu? Jak zmieni się wynik punktu (a), jeśli założymy, że merge zawsze, wykonuje o jedno porównanie mniej, niż zakładaliśmy w punkcie (a)?

Ilość wywołań: $n - 1$

$$\sum_{i=1}^{k} \frac{2^k}{2^i}(2^i - 1) = k2^k - 2^k + 1 = kn - 2^k + 1$$

Zadanie 5

Korzystając z twierdzenia o rekurencji uniwersalnej rozwiąż następujące zależności:

(a) $T(N) = 5T(n/3) + n = \Theta(n^{log_3 5})$, $n = O(n^{log_3 5 - \epsilon})$ dla $\epsilon < 0,46$

(b) $T(N) = 4T(n/2) + n^2 = \Theta(n^2 logn)$, $n^2 = \Theta(n^{log_2 4})$

(c) $T(N) = 9T(n/3) + n^2 = \Theta(n^2 logn)$, $n^2 = \Theta(n^{log_3 9})$

(d) $T(N) = 6T(n/3) + n^2 = \Theta(n^2)$, $n^2 = \Omega(n^{log_3 6})$

(e) $T(N) = 3T(n/3) + n = \Theta(nlogn)$, $n = \Theta(n^{log_3 3})$

(f) $T(N) = 5T(n/2) + n^2 = \Theta(n^{log_2 5})$, $n^2 = O(n^{log_2 5 - \epsilon})$ dla $\epsilon < 0,32$

(g) $T(N) = T(n/2) + 1 = \Theta(logn)$, $1 = \Theta(n^{log_2 1})$