ćwiczenia (licencjat wieczorowy): lista zadań nr 2 *17 marca 2009 r.*

ALGORYTMY I STRUKTURY DANYCH

SORTOWANIE

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. [2 pkt.] Przedstaw w pseudokodzie implementację algorytmu, który pozwoli zrealizować w miejscu dowolną zadaną permutację σ na n–elementowej tablicy $T[0 \ldots n-1]$ (sama permutacja też jest zadana w postaci n–elementowej tablicy $\sigma[0 \ldots n-1]$). Algorytm ma działać w liniowym czasie. Postaraj się, aby wykonywał on co najwyżej $2n$ przepisań elementów.

2. [2 pkt.] Wykaż, że algorytm *sortowania bąbelkowego* na ciągu n–elementowym wykona liniową liczbę zamian elementów (chociaż liczba porównań zawsze będzie kwadratowa) w przypadku, gdy liczba inwersji związanych z każdym elementem jest rzędu $O(1)$.

3. [3 pkt.] Bedziemy uruchamiać algorytm *sortowania bąbelkowego* na n–elementowych ciągach. Rozważmy trzy szczególne przypadki takich ciągów:

   (a) ciag posortowany (0 inwersji);

   (b) ciąg posortowany, w którym zamieniono miejscami parę elementów, pierwszy położony blisko początku a drugi blisko końca tablicy (w przypadku zamiany elementu pierwszego z ostatnim liczba inwersji wyniesie $2n - 3$);

   (c) wewnątrz posortowanego ciągu wyróżniono blok rozmiaru $\lfloor\sqrt{n}\rfloor$, w którym odwrócono kolejność elementów (liczba inwersji wyniesie $\frac{\lfloor\sqrt{n}\rfloor \cdot \lfloor\sqrt{n}-1\rfloor}{2}$).

   Dokonaj takiej modyfikacji w algorytmie sortowania bąbelkowego, która spowoduje, że dla wymienionych przypadków będzie on działał w czasie liniowym $O(n)$.

4. [2 pkt.] Wykaż, że algorytm *sortowania przez wstawianie* na ciągu n–elementowym wykona liniową liczbę porównań i zamian (będzie działał w liniowym czasie), gdy całkowita liczba inwersji w sortowanym ciągu jest rzędu $O(n)$.

5. [3 pkt.] W posortowanym ciągu dokonujemy losowej transpozycji (zamiany pary elementów miejscami). Ile porównań elementów w średnim przypadku wykona algorytm *sortowania przez wstawianie*, gdy uruchomimy go na takim ciągu?