

# Algorytmy i Struktury Danych

## Lista zadań 4 - sortowanie, kopce, drzewa

1. (a) Napisz procedurę, `counting_sort(int t[], int n, int c)`; która posortuje metodą przez zliczanie liczby w tablicy `t[]` względem cyfry `c`.  $c = 0$  oznacza cyfrę jedności,  $c = 1$  cyfrę dziesiątek itd...
2. Posortuj metodą sortowania pozycyjnego liczby: 1, 34, 123, 321, 432, 132, 543, 651, 91, 32, 987, 910, 643, 641, 12, 342, 498, 987, 965, 122, 121, 431, 350.
3. Które z procedur sortujących: (a) insertionsort (przez wstawianie), (b) quicksort (szybkie), (c) heapsort (przez kopcowanie), (d) mergesort (przez złączanie), (e) counting sort (przez zliczanie) (f) radix sort (pozycyjne), (g) bucket sort (kubelkowe) są stabilne. W każdym przypadku uzasadnij stabilność lub znajdź konkretny przykład danych, dla których algorytm nie zachowa się stabilnie.
4. (algorytm Hoare'a) Korzystając funkcji `int partition(int t[], int n)` znanej z algorytmu sortowania szybkiego napisz funkcję `int kty(int t[], int n)`, której wynikiem będzie  $k$ -ty co do wielkości element początkowo nieposortowanej tablicy `t`. Średnia złożoność Twojego algorytmu powinna wynieść  $O(n)$ .
5. Dla jakich danych sortowanie metodą kubelkową ma złożoność  $O(n^2)$ ?
6. Udowodnij, że procedura `build_heap` działa w czasie liniowym.
7. W oparciu o kod procedury `heap_sort` napisz klasę `priority_queue` z metodami `put` i `get_max` oraz `is_empty` i `is_full`. W konstruktorze umieść funkcję `build_heap`.
8. (a) Jaka może być minimalna, a jaka maksymalna ilość kluczy w drzewie czerwono-czarnym o ustalonej czarnej wysokości równej  $h_B$ ?  
(b) Znajdź maksymalną i minimalną wartość stosunku ilości węzłów czerwonych do czarnych w drzewie czerwono-czarnym.
9. (2 pkt.) Do pustego drzewa czerwono-czarnego wstaw kolejno 20 przypadkowych kluczy. Następnie usuń je w tej samej kolejności w jakiej wstawiałeś. Przypadkowymi kluczami są kolejne litery Twojego nazwiska, imienia i adresu. Zadanie wykonujemy na kartce.