

Algorytmy i Struktury Danych

Lista zadań 4 - sortowanie, drzewa, rekurencja

1. Napisz procedurę, `counting_sort(int t[], int n, int c)`; kóra 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. (2 pkt.) Do pustego drzewa czerwono-czarnego wstaw kolejno 20 przypadkowych kluczy. Następnie usuń je w tej samej kolejności w jakiej wstawiałeś. Przypadkowymi liczbami są kolejne litery Twojego nazwiska i imienia.
5. Jaka może być minimalna, a jaka maksymalna ilość kluczy w drzewie czerwono-czarnym ustalonym o czarnej wysokości równej h_B ?
6. Jak jest minimalna a jaka maksymalna liczba kluczy w drzewie czerwono-czarnym o czarnej wysokości h_b .
7. Znajdź maksymalną i minimalną wartość stosunku ilości węzłów czerwonych do czarnych w drzewie czerwono-czarnym?
8. (2 pkt.) Napisz program znajdujący wszystkie ustawienia 8 hetmanów na szachownicy takie, że żaden z nich nie szachuje innego. Oszacuj złożoność twojego rozwiązania.