

Algorytmy i Struktury Danych

Lista zadań 4 - sortowanie i kopce

Uwaga: Zadania z tej listy przygotowujemy w formie pisemnej.

1. Zasymuluj działanie polifazowego mergesorta dla tablicy $\{9, 22, 6, 19, 14, 10, 17, 3, 5\}$. Na każdym etapie sortowania scala się sąsiadujące listy rosnące.
2. a) Czy tablica posortowana malejąco jest kopcem?
b) Czy ciąg $\{23, 17, 14, 6, 13, 10, 1, 5, 7, 12\}$ jest kopcem?
3. Zasymuluj działanie `buildheap` dla $t[] = \{1, 2, 3, 4, 5, 6, 7\}$. Zapisz na kartce wygląd tablicy/kopca po każdym wywołaniu procedury `przesiej`.
4. Zilustruj działanie procedury `buildheap` dla ciągu $\{5, 3, 17, 10, 84, 19, 6, 22, 9\}$. Narysuj na kartce wygląd tablicy/kopca po każdym wywołaniu procedury `przesiej`.
5. 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`.
6. Udowodnij, że procedura `build_heap` działa w czasie liniowym.
7. Udowodnij, że wysokość kopca n -elementowego wynosi $\lceil \log_2 n \rceil$.
8. Wykaż, że pesymistyczna złożoność `quicksort` wynosi $O(n^2)$.
9. Zakładając, że $t[] = \{7, 6, 5, 4, 3, 2, 1\}$ i stosując algorytmy sortujące ściśle wg procedur z pliku `sorty.cc`:
(a) zasymuluj działanie `partition(t, 7)`.
(b) Zasymuluj działanie `partition(t, 7)` w przypadku gdyby pivotem zamiast $t[n/2]$ było $t[0]$.
10. Zakładając, że $t1[] = \{1, 2, 3, 4, 5, 6, 7\}$ oraz $t2[] = \{7, 6, 5, 4, 3, 2, 1\}$ oblicz ile porównań (między elementami tablicy) wykona: (a) `insertion_sort(t1)`, (b) `insertion_sort(t2)`.