

LABORATORIUM NR 2

SORTOWANIE PRZEZ KOPCOWANIE

ZADANIE AiSD.2.1

Sprawdź, czy ciąg (23, 17, 14, 6, 13, 10, 1, 5, 7, 12) spełnia własność kopca typu MAX?

ZADANIE AiSD.2.2

Zademonstruj kolejne kroki działania procedury budującej kopiec typu MAX z elementów tablicy $A = [28, 6, 11, 12, 17, 8, 7, 18, 12, 14, 23]$. Następnie zilustruj kilka obrotów pętli sortującej przy sortowaniu kopcowym `Heapsort`. Aby wskazywać, które węzły kopca są ze sobą zamieniane, użyj drzewiastej reprezentacji kopca.

ZADANIE AiSD.2.3 (4.5 pkt.)

- Zaimplementuj omawiany na wykładzie algorytm sortowania (porządek nierosnący) przez kopcowanie. (3.5 pkt.)
- Zmodyfikuj procedurę `Heapify` tak, aby używała iteracji zamiast rekursji. (1 pkt)

Specyfikacja wejścia/wyjścia.

Wejście. Liczby (całkowite) zapisane w kolejnych wierszach pliku tekstowego.

Wyjście. Posortowane liczby z pliku wejściowego zapisane w kolejnych wierszach pliku wyjściowego.

ZADANIE AiSD.2.4 (4.5 pkt.)

Zaimplementuj algorytm sortowania przez kopcowanie, który sortuje (porządek nierosnący) tylko wskazany zakres ciągu n liczb całkowitych.

Specyfikacja wejścia/wyjścia.

Wejście. $n + 2$ liczb zapisanych w kolejnych wierszach pliku tekstowego, z czego dwie pierwsze x i y spełniają zależność $1 \leq x \leq y \leq n$, a kolejne są elementami tablicy A .

Wyjście. Posortowany ciąg liczb $A[x - 1], A[x], \dots, A[y - 1]$ zapisany w kolejnych wierszach pliku wyjściowego.

ZADANIE AiSD.2.5 (0.5 pkt., rozwiązanie/zestawienie na kartce)

Oszacuj czasy działania algorytmu sortowania przez kopcowanie dla tablicy A o długości n , w której (a) wszystkie elementy są takie same, (b) są posortowane malejąco, (c) są posortowane rosnąco. Następnie, w oparciu o rozwiązanie zadania AiSD.2.3 (AiSD.2.4), dokonaj eksperymentalnego pomiaru czasu dla zaproponowanych oszacowań (patrz zadanie AiSD.1.1).

