

Kopiec binarny i kolejka priorytetowa

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

Uwaga: Słowa „dany”, „zadany”, „podany”, „wybrany” itd. w kontekście parametrów (zmiennych) oznacza parametr zadany przez użytkownika (a nie na stałe, przez programistę), a implementacja wykorzystująca taki parametr powinna obsługiwać jego różne wartości.

1 Zadanie nr 1 – budowa kolejki

Zaimplementuj trzy wersje kolejki priorytetowej (z operacjami: wstawiania, usuwania, wyszukiwania i generacji kolejki – wszystkie operacje efektywnymi algorytmami):

- naiwną (tablica) nieposortowaną,
- naiwną (tablica) posortowaną,
- z wykorzystaniem kopca binarnego.

Dopuszczaj wersje priorytetu typu minimum i maksimum.

2 Zadanie nr 2 – kodowanie

- Zaimplementuj algorytm generujący kod (kodowanie) Huffmana dla zadanego ciągu słów (tekstu) i zadanego ciągu prawdopodobieństw wystąpienia słów.
- Zaimplementuj metodę szacującą prawdopodobieństwa występowania słów w zadanym tekście.
- Uzupełnij metodę o „czyszczenie” tekstu z nieobsługiwanych znaków (pozostaw tylko litery abecadła).
- Zaimplementuj metodę tworzącą kod Huffmana dla zadanego tekstu (po automatycznym oczyszczeniu i automatycznej generacji prawdopodobieństw).

3 Zadanie nr 3 – sortowanie

- Zaimplementuj algorytm sortowania przez kopcowanie.
- Proces sortowania przedstaw graficznie (np. korzystając z wykresu kolumnowego `matplotlib`). Wizualizacja ma wyświetlać kolejne kroki algorytmu, gdzie kolejny krok zachodzi po wykonaniu elementarnej operacji (porównywania liczb, zmiany położenia liczb w ciągu, itd.).
- Wykonaj analogiczną wizualizację dla algorytmu sortowania szybkiego (quicksort). Obie wizualizacje przedstaw (wyświetl) równolegle (podobnie jak na poprzedniej liście).