Bit Algo
START

Sprawy organizacyjne

- zajęcia regularnie co tydzień: poniedziałek 16:15-17:50 (3.27), środa 16:15-17:50 (?)

- https://www.facebook.com/groups/bitalgoagh/

Pon 09.03.2020
Merge i Quick - zadania

Założenia

- Rozumiecie MergeSorta
- Rozumiecie kopce min i max
- Rozumiecie QuickSorta

Zadania
MergeSort i QuickSort

Zadanie 1 [Algorytm]

3. Mamy serię pojemników z wodą, połączonych (każdy z każdym) rurami. Pojemniki maja kształty prostokątów (2d), rury nie maja objętości (powierzchni). Każdy pojemnik opisany jest przez współrzędne lewego górnego rogu i prawego dolnego rogu. Wiemy, ze do pojemników nalano A wody (oczywiście woda rurami spłynęła do najniższych pojemników). Obliczyć ile pojemników zostało w pełni zalanych.

Zadanie 2 [Algorytm, implementacja]

Dana jest tablica zawierająca liczby naturalne. Proszę zaimplementować funkcję odpowiadającą na pytanie czy w tablicy jest para sumująca się do jakiejś liczby x. Funkcja powinna być jak najszybsza.

findPair(arr, x) -> bool.

https://github.com/BIT-LGO-START/Semestr2-2019-2020/blob/master/findPair.py

Zadanie 3 [Algorytm]

Zaproponuj algorytm scalający k posortowanych tablic w jedną posortowaną tablicę. Łączna liczba elementów we wszystkich tablicach wynosi n. Algorytm powinien najlepiej działać w czasie O(n*log(k)).

https://github.com/BIT-LGO-START/Semestr2-2019-2020/blob/master/merge_k_sorted_lists.py

Zadanie 4 [Algorytm]

Dana jest n-elementowa tablica A zawierająca liczby naturalne (potencjalnie bardzo duże). Wiadomo, że tablica A powstała w dwóch krokach. Najpierw wygenerowano losowo (z nieznanym rozkładem) n różnych liczb nieparzystych i posortowano je rosnąco. Następnie wybrano losowo ceil(log n) elementów powstałej tablicy i zamieniono je na losowo wybrane liczby parzyste. Zaproponuj (bez implementacji!) algorytm sortowania tak powstałych danych. Algorytm powinien być możliwie jak najszybszy. Proszę oszacować i podać jego złożoność czasową.

Zadanie 5 [Algorytm]

Zaimplementuj funkcję average_score(arr, n, lowest, highest). Funkcja ta przyjmuje na wejściu tablicę n liczb rzeczywistych (ich rozkład nie jest znany, ale wszystkie są parami różne) i zwraca średnią wartość podanych liczb po odrzuceniu lowest najmniejszych oraz highest największych. *Zaimplementowana funkcja* powinna być możliwie jak najszybsza. Oszacuj jej złożoność czasową (oraz bardzo krótko uzasadnić to oszacowanie).

Ale my skupimy się tylko na algorytmie :D

Zadanie 6 [Algorytm]

Dana jest tablica zawierająca liczby rzeczywiste. Różnych wartości w tablicy jest tylko log(n), gdzie n to rozmiar tablicy. Proszę zaproponować algorytm sortujący tablicę w czasie O(nlog(log(n)). Wskazówka: problem da się rozwiązać wykorzystując algorytm wyszukiwania binarnego i dodatkową tablicę o rozmiarze log(n).

Zadanie 7 [Algorytm, Implementacja]

Proszę zaimplementować algorytm QuickSort, tak aby głębokość stosu rekursji nie przekraczała O(log(n)).

https://github.com/BIT-LGO-START/Semestr2-2019-2020/blob/master/NotDeepQuickSort.py

Bit Algo
START