

# Alгоритмы и структуры данных

## Lista 3

### Zadanie 1.

Podaj algorytm scalający  $k$  posortowanych list tak aby powstała jedna posortowana lista nb (liczba wszystkich elementów na listach to  $n$ ) działający w czasie  $O(n \log k)$ .

### Zadanie 2.

Zdefiniujmy algorytm  $k$ -MergeSort jako uogólnienie algorytmu sortowania przez scalanie. Różni się od omawianego na wykładzie algorytmu sortowania przez scalanie tym, że dzieli sortowaną tablicę rekurencyjnie na  $k$  równych części (zakładamy, że liczba elementów w tablicy jest potęgą  $k$  ( $n = k^l$ )). Używając wyniku z zadania 1 proszę wykazać dla jakiego  $k$  algorytm ma najmniejszą asymptotyczną złożoność obliczeniową liczby porównań (górne ograniczenie  $O()$ ).

### Zadanie 3.

Założmy że tablica  $A = [a_1, \dots, a_n]$  jest do pewnego momentu  $k$  posortowana malejąco i dalej rosnąco (tzn. dla  $\forall i \leq k$   $a_i > a_{i+1}$  oraz dla  $\forall i \geq k$   $a_i < a_{i+1}$ ). Zaprojektuj algorytm znajdujący minimalny element w tablicy  $A$ , którego złożoność obliczeniowa będzie wynosić  $O(\log n)$ . Udowodnij poprawność działania zaproponowanego algorytmu.

### Zadanie 4.\*

Zastąpienie użycia QuickSort'a, dla tablic małych rozmiarów, algorytmem InsertionSort jest częstym sposobem na zwiększenie efektywności algorytmu rozwiązującego problem sortowania. Pokaż, że jeśli zmodyfikujesz bazowy przypadek rekurencji w algorytmie QuickSort w taki sposób, że dla tablic o  $\leq k$  elementach wywoływany będzie InsertionSort (zamiast rekurencyjnego wywołania QuickSort'a) to wartość oczekiwana liczby porównań będzie wynosić  $\Theta(nk + n \log \frac{n}{k})$ . Jakie  $k$  należy wybrać aby zminimalizować tą złożoność?

### Zadanie 5.

Założmy, że masz do wyboru jeden z trzech algorytmów rozwiązujących postawiony Ci problem wielkości  $n$ :

- Algorytm A: rozwiązuje problem dzieląc go rekurencyjnie na 5 pod-problemów o połowę mniejszych i scalając ich rozwiązania w czasie  $\Theta(n \log(n))$ .
- Algorytm B: rozwiązuje problem dzieląc go rekurencyjnie na 2 pod-problemy rozmiaru  $n - 1$  i scala ich rozwiązania w czasie stałym.
- Algorytm C: rozwiązuje problem dzieląc go rekurencyjnie na 9 pod-problemów rozmiaru  $\frac{n}{3}$  i scalając ich rozwiązania w czasie  $\Theta(n^2)$ .

Jaka jest złożoność obliczeniowa tych algorytmów? Który z nich byś wybrał? Odpowiedź uzasadnij.

### Zadanie 6.

Powiedzmy, że masz do wykonania  $n$  zadań, gdzie każde z nich wymaga  $t_j$  minut pracy. Chcesz wykonać wszystkie zadania maksymalizując zadowolenie przełożonego poprzez minimalizację średniego czasu zakończenia każdego zadania. Uzasadnij w jakiej kolejności powinieneś wykonywać zadania.

### Zadanie 7.

Stwórz algorytm znajdujący najczęściej powtarzający się element w  $n$  elementowej tablicy (unikając sortowania tablicy), mający złożoność  $O(n \log n)$  (zakładamy, że ten element powtarza się ponad  $\frac{n}{2}$  razy).

### Zadanie 8.

Doktor Freud ma wahania nastrojów, które zapisuje sobie ilustrując nastrój danego dnia nieujemną liczbą całkowitą. Po  $n$  dniach zgromadził tablicę  $n$  liczb opisujących swój nastrój i postanowił znaleźć (spójny) przedział czasu (dni), w których był najszczęśliwszy. Doktor Freud zdefiniował sobie szczęśliwość przedziału czasu jako sumę wartości występujących w dniach tego przedziału przemnożony przez

najmniejszą wartość występującą w zadanym przedziale. Stwórz algorytm D&C znajdujący najszcześniejszy przedział w złożoności  $O(n \log n)$

**Zadanie 9.**

Wykaż, że nie istnieje algorytm sortujący, który działa w czasie liniowym dla co najmniej połowy z  $n!$  możliwych danych wejściowych długości  $n$ . Czy odpowiedź ulegnie zmianie jeśli zapytamy o ułamek  $\frac{1}{n}$  lub  $\frac{1}{2^n}$  wszystkich permutacji?

**Zadanie 10.**

Zaprojektuj algorytm, który sortuje  $n$  liczb całkowitych z przedziału od 1 do  $n^2$  w czasie  $O(n)$ .