Ćwiczenia 8

Zadanie 1, kolokwium 1, 2017/18

W liczbowym, różnowartościowym ciągu $(a_1, a_2, \ldots, a_n)$, $n > 2$, element a_i, $1 < i < n$, nazywamy lokalnym ekstremum, gdy jest mniejszy lub większy od obu sąsiadów, tzn. albo $a_{i-1} > a_i < a_{i+1}$, albo $a_{i-1} < a_i > a_{i+1}$.

(a) Udowodnij, że każdy algorytm sortujący przez porównania 4-elementowe ciągi z co najwyżej jednym lokalnym ekstremum wymaga wykonania w pesymistycznym przypadku co najmniej 4 porównań.

(b) Zaproponuj algorytm sortowania 4-elementowych ciągów z co najwyżej jednym lokalnym ekstremum za pomocą co najwyżej 4 porównań.

(c) Zaproponuj algorytm, asymptotycznie optymalny ze względu na liczbę porównań, sotrujący ciągi o co najwyżej k lokalnych ekstremach dla zadanego k, $0 < k < n$. Dowiedź optymalności swojego rozwiązania.

Zadanie 1, kolokwium 1, 2021/22

Powiemy, że zbiór n liczb całkowitych jest prawie gęsty, jeśli zawiera podzbiór o rozmiarze większym niż $n/3$, w którym różnica pomiędzy największym i najmniejszym elementem jest mniejsza od n. Taki podzbiór nazywamy świadectwem. Dana jest dodatnia liczba całkowita n oraz n-elementowy zbiór liczb całkowitych S. Zaproponuj algorytm, który w czasie liniowym sprawdzi, czy S jest prawie gęsty.

Uwaga: połowę punktów uzyskasz za algorytm, który w czasie liniowym sprawdza, czy wskazany, dowolny element z S należy do jakiegoś świadectwa.

Zadanie 3, kolokwium 1, 2021/22

Dla dodatniej liczby całkowitej n kratownicą M_n nazywamy skierowany graf (V, E) bez pętli, w którym $V = \{(x, y) : x = 0, 1, \ldots, n \text{ oraz } y = 0, 1, ..., n\}$ i

$$E = \{(x, y) \to (x', y') : 0 \leqslant x' - x \leqslant 1 \text{ oraz } 0 \leqslant y' - y \leqslant 1\}.$$

Wierzchołki grafu M_n pomalowano na biało lub czarno. Białą ścieżką nazwiemy każdą ścieżkę, na której wszystkie wierzchołki są białe. Dane są liczba całkowita $n > 0$, nieujemna liczba całkowita $m \leqslant (n + 1)^2$ oraz m różnych, białych wierzchołków $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$. Pozostałe wierzchołki są czarne. Zaproponuj wydajny (czasowo i pamięciowo) algorytm, który obliczy liczbę wszystkich białych ścieżek z wierzchołka $(0, 0)$ do wierzchołka (n, n).