Ćwiczenia 5

3.4 Powiemy, że dwa słowa są podobne wtedy i tylko wtedy, gdy zawierają jednakowe liczby wystąpień tych samych znaków. Danych jest n słów nad alfabetem m-znakowym $\{1, 2, \ldots, m\}$. Zaproponuj algorytm, który stwierdza, ile jest wśród nich różnych klas słów podobnych. Twój algorytm powinien działać w czasie $O(R + m)$, gdzie R jest sumą długości wszystkich słów.

4.1 W tym zadaniu analizujemy algorytm QuickSort, w którym za element dzielący wybiera się medianę z $(a[l],\ a[\lfloor (l+r)/2 \rfloor + 1],\ a[r])$ (zobacz wykład 3). Rozważmy permutację

$$K_{2k} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & \ldots & k-2 & k-1 & k & k+1 & k+2 & k+3 & \ldots & 2k-1 & 2k \\ 1 & k+1 & 3 & k+3 & 5 & \ldots & 2k-3 & k-1 & 2k-1 & 2 & 4 & 6 & \ldots & 2k-2 & 2k \end{pmatrix}.$$

Udowodnij, że dla permutacji K_n, gdzie n jest dodatnią liczbą całkowitą podzielną przez 4, QuickSort działa w czasie $\Omega(n^2)$.

4.2 Niech n będzie liczbą całkowitą większą od 1, a

$$X = \{(x, y) : x = 0, 1, \ldots, n-1, y = 0, 1, 2, 3, 4\}$$

będzie zbiorem punktów na płaszczyźnie. Danych jest n różnych prostych, z których każda przechodzi przez dwa różne punkty ze zbioru X, różniące się zawsze pierwszą współrzędną. Każda prosta zadana jest przez parę punktów $\{(x_1, y_1), (x_2, y_2)\}$, gdzie $x_1 < x_2$. Zaprojektuj algorytm, który w czasie liniowym posortuje wszystkie proste niemalejąco względem ich kątów nachylenia do osi OX.

4.4 Dla dodatniej liczby całkowitej d, $d > 1$, d-kopcem typu MIN, nazywamy zupełne drzewo d-arne z kluczami rozmieszczonymi w porządku kopcowym typu MIN. Zaproponuj wydajną implementację d-kopca w tablicy i dokonaj analizy złożoności obliczeniowych operacji kolejki priorytetowej: Ini, Min, DeleteMin, Insert i DecreasKey. W jaki sposób dobrać d, żeby dostać jak najszybszą implementację algorytmu Dijkstry, uwzględniającą liczbę krawędzi w grafie.

Do samodzielnej pracy

4.3 Podaj permutację liczb 1, 2, ..., 7, dla której algorytm HeapSort (w wersji z wykładu) wykona największą liczbę porównań.

Uwaga: należy wziąć pod uwagę obie fazy algorytmu budowę kopca i właściwe sortowanie.