# Egzamin z ASD
# 30.01.2018

## Zadanie 1 [8 punktów]

W kostce $[0,n]^3$, $n > 10$, danych jest $n$ różnych odcinków równoległych do osi układu współrzędnych i o końcach w punktach o współrzędnych całkowitych, każdy o długości 10. Zaproponuj efektywny algorytm obliczający liczbę punktów o współrzędnych całkowitych należących do co najmniej dwóch różnych odcinków.

## Zadanie 2 [16 punktów]

Niech $n$ będzie liczbą całkowitą większą od 2 i niech $J$ będzie rodziną co najwyżej $n$ różnych, domkniętych przedziałów liczb całkowitych zawartych w przedziale $[1,n]$. Grafem $G(n,J)$ nazywamy graf $(\{1,2, \ldots, n\},\{i\text{-}j$: **istnieje** przedział w $J$, do którego wpadają obie liczby (oba wierzchołki) $i$ oraz $j\})$.

a) [2 punkty] Ile jest różnych drzew BFS o korzeniu w wierzchołku 1 w grafie $G(8,J)$ dla $J = \{[1,4],[3,6],[5,8]\}$?
b) [2 punkty] Ile jest różnych drzew DFS o korzeniu w wierzchołku 1 w grafie $G(6,J)$ dla $J = \{[1,4],[3,6]\}$?
c) [7 punktów] Zaprojektuj efektywny algorytm, który dla danej liczby całkowitej $n > 2$ oraz rodziny co najwyżej $n$ różnych przedziałów $J$ zawartych w przedziale $[1,n]$ obliczy wysokość BFS drzewa w grafie $G(n,J)$, o korzeniu w wierzchołku 1.
d) [5 punktów] Zaprojektuj efektywny algorytm, który dla danej liczby całkowitej $n > 2$ oraz rodziny co najwyżej $n$ różnych przedziałów $J$ zawartych w przedziale $[1,n]$, obliczy liczbę dwuspójnych składowych w grafie $G(n,J)$.

Uwaga: na potrzeby tego zadania dwa drzewa przeszukiwania różnią się wtedy, gdy istnieje wierzchołek, który w obu drzewach ma różnych ojców.

## Zadanie 3 [7 punktów]

Niech $x$ będzie słowem binarnym o długości co najmniej 2 i zawierającym co najmniej jedno 0 (zero) oraz co najmniej jedną 1 (jedynkę). Zaprojektuj efektywny algorytm, który w słowie binarnym $x$ znajduje dwa podsłowa o maksymalnej długości, które różnią na każdej pozycji.

## Zadanie 4 [9 punktów]

Zaprojektuj strukturę danych, która umożliwia wydajne wykonywanie następujących operacji na dynamicznie zmieniającym się ciągu liczbowym $a = \langle a_1, a_2, \ldots, a_n \rangle$:

Ini($a$):: $a$ := <pusty ciąg>; //operacja wykonywana tylko raz na początku
Insert($a,a,i$):: wstaw element $a$ na pozycję $i$ w ciągu $a$, $1 \leq i \leq |a|+1$
Delete($a,i$):: usuń $i$-ty element z ciągu $a$, $1 \leq i \leq |a|$
Add($a,e,i,j$):: do każdego z elementów podciągu $a[i..j]$ dodaj $e$, $1 \leq i \leq j \leq |a|$
MaxInc($a$):: podaj długość najdłuższego **spójnego** podciągu rosnącego w ciągu α

**Uwaga:** uzasadnij poprawność swoich rozwiązań i dokonaj analizy złożoności obliczeniowej zaproponowanych algorytmów.