

# Zadania kwalifikacyjne – uczenie maszynowe

Wakacyjne Warsztaty Wielodyscyplinarne

Krzysztof Zając

Uwaga: nie trzeba robić obu zadań. Zrób ile potrafisz – to, czy zrobisz jedno, czy dwa, i czy łatwą, czy trudną wersję pierwszego zadania, może mieć wpływ na kwalifikację gdy będzie dużo zgłoszeń.

Rozwiązania wyślij na adres [krzysztof.zajac2@gmail.com](mailto:krzysztof.zajac2@gmail.com) (w przypadku jakichś problemów czy pytań też pisz).

## Zadanie 1

Wybierz jedną wersję tego zadania - trudniejszą lub łatwiejszą.

### Wersja łatwiejsza - dopasowanie okręgu do punktów

Danych jest  $n$  punktów, o których wiemy, że leżą (w przybliżeniu) na pewnym okręgu. Napisz (w swoim ulubionym języku programowania) program, który wczytuje punkty i znajduje ten okrąg. Program powinien sobie radzić nawet, jeśli niektóre punkty nie leżą dokładnie na tym okręgu.

#### Dane wejściowe

Program powinien wczytać ze standardowego wejścia liczbę naturalną  $n$ , oznaczającą liczbę punktów na wejściu. Następnie powinien wczytać  $n$  par liczb rzeczywistych oznaczających współrzędne (kolejno  $x$  i  $y$ ) poszczególnych punktów.

#### Dane wyjściowe

Program powinien wypisać na standardowe wyjście trzy liczby: współrzędne  $x$  i  $y$  środka okręgu a następnie jego promień.

## Przykład

Program, otrzymawszy następujące dane wejściowe: 4 0 1 1 0 0 -1 -1 0 (cztery punkty, o współrzędnych odpowiednio  $(0, 1)$ ,  $(1, 0)$ ,  $(0, -1)$ ,  $(-1, 0)$ ) powinien wypisać następujące liczby: 0 0 1 (okrąg o środku w punkcie  $(0, 0)$  i promieniu 1).

## Wersja trudniejsza - dopasowanie elipsy do punktów

Danych jest  $n$  punktów, o których wiemy, że leżą (w przybliżeniu) na pewnej elipsie. Napisz (w swoim ulubionym języku programowania) program, który wczytuje punkty i znajduje tę elipsę. Program powinien sobie radzić nawet, jeśli niektóre punkty nie leżą dokładnie na tej elipsie.

### Dane wejściowe

Program powinien wczytać ze standardowego wejścia liczbę naturalną  $n$ , oznaczającą liczbę punktów na wejściu. Następnie powinien wczytać  $n$  par liczb rzeczywistych oznaczających współrzędne (kolejno  $x$  i  $y$ ) poszczególnych punktów.

### Dane wyjściowe

Program powinien wypisać na standardowe wyjście sześć liczb: współrzędne  $x$  i  $y$  jednego ogniska elipsy, następnie współrzędne  $x$  i  $y$  drugiego ogniska elipsy, a następnie (w dowolnej kolejności) długości obu pól.osi.

## Przykład

Program, otrzymawszy następujące dane wejściowe: 4 0 1 2 0 0 -1 -2 0 (cztery punkty, o współrzędnych odpowiednio  $(0, 1)$ ,  $(2, 0)$ ,  $(0, -1)$ ,  $(-2, 0)$ ) powinien wypisać następujące liczby: 1.73205 0 -1.73205 0 1 2 (elipsa o ogniskach w punktach  $(1.73205, 0)$  i  $(-1.73205, 0)$  i półosiach długości 1 i 2).

## Zadanie 2

Jak byś przy użyciu algorytmu  $k$  najbliższych sąsiadów (por. Wikipedia: [http://pl.wikipedia.org/wiki/K-najbli%C5%BCszych\\_s%C4%85siad%C3%B3w](http://pl.wikipedia.org/wiki/K-najbli%C5%BCszych_s%C4%85siad%C3%B3w)) napisał program, który dostaje na wejściu czasownik (być może odmieniony) i wypisuje na wyjściu jego formę w bezokoliczniku?

Jak, na potrzeby algorytmu, zdefiniować, które słowo jest *najbliższe* danemu?

Możesz założyć, że masz do dyspozycji duży słownik postaci:

- *analizowałbym* → *analizować*
- *bierzesz* → *brać*
- *głaszczesz* → *głaskać*

Oczywiście algorytm powinien sobie radzić ze słowami spoza słownika.

Rozwiązaniem może być opis algorytmu – nie trzeba (choć można) pisać programu.