Cvičenie 03: Pravdepodobnosť a triedenie

Efektívne Algoritmy a Dátové Štruktúry
FMFI UK

Opakovanie: stredná hodnota

Definícia strednej hodnoty:

$$E[X] = \sum_x x \cdot \Pr(X = x)$$

Linearita strednej hodnoty:

$$E[X + Y] = E[X] + E[Y]$$

Stredná hodnota (1)

Niekto háďže n-krát mincou. $\Pr(X = 1) = \frac{1}{2}$, $\Pr(X = 0) = \frac{1}{2}$, ja tipujem náhodne.

Aký je očakávaný počet hodov, v ktorých sa trafím?

Stredná hodnota (2)

Niekto háďže n-krát mincou. $\Pr(X = 1) = \frac{1}{3}$, $\Pr(X = 0) = \frac{2}{3}$, ja tipujem náhodne.

Aký je očakávaný počet hodov, v ktorých sa trafím?

Stredná hodnota (3)

Postupne sa strieda 0 a 1, ja tipujem náhodne. Aký je očakávaný počet hodov, v ktorých sa trafím?

Stredná hodnota (4)

Ako chcem tipovať, ak viem, že ide o druhý prípad?

$\Pr(X = 1) = \frac{1}{3}, \Pr(X = 0) = \frac{2}{3}$

Aplikácie triedenia (1)

Dané je pole čísiel. Pre každé číslo vypíšte, koľkokrát sa v poli vyskytuje.

Aplikácie triedenia (2)

Anagram je slovo vytvorené preskupením písmen iného slova (lampa $\longrightarrow$ palma).

Máme zoznam všetkých existujúcich slov. Zistite, ktoré slovo má najviac anagramov.

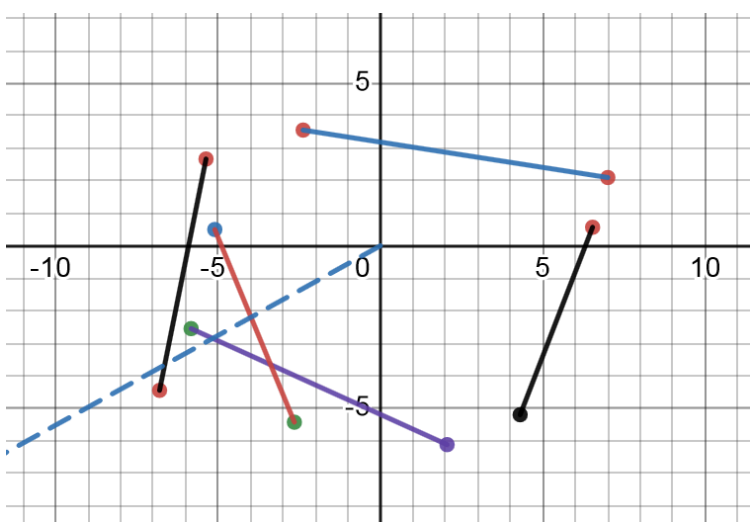
Aplikácie triedenia (3)

Máme n prednášok, o každej vieme jej začiatok a koniec. Chceme zistiť, koľko najviac prednášok beží súbežne.

Aplikácie triedenia (4)

Lukostrelec strieľa z bodu $[0, 0]$ ľubovoľným smerom. Koľko najviac terčov vie zasiahnuť jedným šípom?

(Šíp po náraze do terča pokračuje ďalej rovnakým smerom.)



Python: `x.sort()` vs `sorted(x)`

```
arr = [ 3, 2, 1 ]
arr.sort()

# arr = [ 1, 2, 3 ]
```

`x.sort()` utriedi existujúce pole

```
arr = [ 3, 2, 1 ]
arr2 = sorted(arr)

# arr =  [ 3, 2, 1 ]
# arr2 = [ 1, 2, 3 ]
```

`sorted(x)` vytvorí kópiu poľa a utriedi ju

Python: zostupné (↘) triedenie

```
arr = [...]

arr.sort(reverse=True)

arr = sorted(arr, reverse=True)
```

Python: triedenie podľa vlastného kľúča

```
arr = ["c", "B", "a"]
arr.sort()
# ["B", "a", "c"]

arr.sort(key=str.lower)
# ["a", "B", "c"]
```

`key` môže byť ľubovoľná funkcia, ktorá akceptuje prvok poľa a vráti hodnotu, podľa ktorej sa má triediť.

Python: Vlastná porovnávacia funkcia

```
from functools import cmp_to_key

def custom_func(item1, item2):
    ...

arr.sort(key=cmp_to_key(custom_func))
```

Funkcia `custom_func` vracia číslo, ktoré porovnáva tieto dva prvky:

- < 0: `item1` má byť triedené **pred** `item2`
- > 0: `item1` má byť triedené **za** `item2`
- $= 0$: `item1` má byť triedené **rovnako** ako `item2`

Pomôcka na zapamätanie: predvolená porovnávacia funkcia je `item1 - item2`.