

Domáca úloha 2

Maximálny počet bodov: 25

Odovzdanie: **Do 20. 10. 20:00**

Pokyny

Domácu úlohu odovzdávajte do odevzdávareny:

https://is.muni.cz/auth/el/1433/podzim2018/IB111/ode/skupina_34_bojnak/domaca_uloha_2/

Ustrážte si termín odovzdania, odevzdávarena sa v danom čase zatvára a úlohu potom nebude možné odovzdať.

Prosím, dávajte si pozor na odpisovanie, ktoré sa tvrdo trestá. Môžete si pomáhať internetom, akurát pokiaľ nejaký kód odniekiaľ skopírujete, budem požadovať, aby ste mu rozumeli (môžem od vás požadovať, aby ste svoj kód vysvetlili). Odporúčam vám ale, aby ste úlohu pre vaše dobro robili sami, tak sa najlepšie naučíte programovať.

Znovu si nezapudnite prečítať pokyny na adrese <https://www.fi.muni.cz/IB111/?p=dupokyny> a riadiť sa nimi pri odovzdávaní.

V prípade akýchkoľvek nejasností ma neváhajte kedykoľvek kontaktovať.

Opäť prajem veľa šťastia :)

Zadanie

Rozdeľovanie sladkostí

Majme n sladkostí, ktoré chceme rozdeliť medzi m detí. Rozhodli sme sa to medzi ne rozhodiť náhodne (aby sme boli škodoradostní a mohli sa smiať na det'och, čo nedostanú náhodou nič). Budeme to robiť pomocou klasickej kocky, kde výsledok hodu je náhodné celé číslo od 1 do 6.

Úloha 1: Simulátor rozdeľovania sladkostí

Naprogramujte funkciu `game(sweet_number, children, output = True)`, ktorá bude simulovať náhodné rozdeľovanie sladkostí. Rozdeľovanie má nasledujúce pravidlá:

- V hre je n sladkostí a m detí (deti sú pomenované číslami 1 až m)
- Začína dieťa s menom 1, pokračuje sa postupne dieťaťom 2, 3... n
- Každé dieťa hádže kockou (1-6), keď hodí nejaké dieťa 6, tak hádže ešte raz
- Každé dieťa získa v každom kole buď toľko sladkostí, koľko je súčet jeho hodov (čiže ak v jednom kole hodilo dieťa 6, následne zas 6 a následne 3, tak dostane 15 sladkostí), alebo nedostane nič, ak počet sladkostí je menší ako súčet jeho hodov a následne pokračuje ďalšie dieťa
- Hra končí, keď dojdú všetky sladkosti
- Pokiaľ počet sladkostí, alebo detí je menšie ako 1, skončíte funkciu hneď na začiatku a vypíšete na výstup nejakú primeranú hlášku
- Ak už každé dieťa hádzalo, hádže sa od začiatku
- Pokiaľ je vo funkcii `output` nastavený na `true`, bude priebežný stav hry (tak ako je to na ukázkovom výstupe programu nižšie) vypisovaný na výstup, inak sa nebude vypisovať nič, funkcia bude iba počítat'.

Ukážkový výstup programu:

```
>>> game(10,3)
1. Child: 6 1 -> Gets: 7 -> Remaining sweets: 3
2. Child: 1 -> Gets: 1 -> Remaining sweets: 2
3. Child: 1 -> Gets: 1 -> Remaining sweets: 1
1. Child: 5 -> Gets: 0 -> Remaining sweets: 1
2. Child: 5 -> Gets: 0 -> Remaining sweets: 1
3. Child: 2 -> Gets: 0 -> Remaining sweets: 1
1. Child: 6 3 -> Gets: 0 -> Remaining sweets: 1
2. Child: 3 -> Gets: 0 -> Remaining sweets: 1
3. Child: 6 1 -> Gets: 0 -> Remaining sweets: 1
1. Child: 1 -> Gets: 1 -> Remaining sweets: 0
All sweets distributed to children in 10. round
```

Úloha 2 – Analýza priemerného počtu ťahov

Vytvorte funkciu `game_analysis(sweet_number, children, count)`, ktorá ako typ vráti float a ktorá analyzuje, koľko priemerne počtu ťahov rozdeľovania je potrebné pre zadaný počet sladkostí, počet detí a počet opakovaní hry.

Úloha 3 – Výpis priemerných počtov ťahov

Pomocou funkcie z predchádzajúcej úlohy vytvorte funkciu `game_average_length(children, count)`, ktorá pre zadaný počet detí a počet opakovaní hry vypíše na výstup jednotlivé výsledky analýz postupne pre 1 až 50 sladkostí.

Poznámka 1: Snažte sa dodržiavať programovací štýl PEP8, ktorý je štandardným Python programovacím štýlom. Prečítajte si napríklad článok [pep8](#) a riadiť sa ním. Body za to strhávať nebudem, od budúcich úloh sa to ale môže zmeniť, preto pozor na to.

Poznámka: Nebojte sa vytvárať vlastné funkcie a vyhýbať sa tak opakovaniu kódu. Základná kostra úlohy vyzerá takto (namiesto `pass` píšete svoju implementáciu úlohy, toto kľúčové slovo sa používa, keď je funkcia prázdna):

```
from random import randint, random

def game(length, output = True):
    pass

def game_analysis(length, count):
    pass

def game_average_length(count):
    pass
```