

## 6. Generování náhodných čísel a testování generátorů

### Zadání:

Tento úkol bude poněkud kreativnější charakteru. Vaším úkolem je vytvořit vlastní generátor semínka do pseudonáhodných algoritmů. Jazyk Python umí sbírat přes ovladače hardwarových zařízení různá fyzická a fyzikální data. Můžete i sbírat data z historie prohlížeče, snímání pohybu myši, vyzvání uživatele zadat náhodné úhozy do klávesnice a jiná unikátní data uživatelů.

### Řešení:

Na řešení vytvoření generátoru náhodných čísel jsem využil informace o počítači a času. Mezi informace, které jsem si o počítači zjistil bylo za pomoci knihovny platform. Tyto informace jsem poté za pomoci funkce `cisloZeStringu` převedl do čísla.

```
def cisloZeStringu(string: str):  
    cislo = 0  
    for char in string:  
        cislo += ord(char)  
    return cislo
```

Semínko nakonec vznikalo z využití cpu uživatelem následně násobením a dělením informací o systému a aktuálnímu času. Pro větší znárodnění jsem přidal parametr předešlé, kam jsem házel součet již předešle vytvořených čísel to se ve výsledku ukázalo jako velice dobrý krok, jelikož výsledný generátor byl o hodně vylepšen. Semínko se nakonec vracelo v určitém rozpětí daném parametrem `size`, a nakonec zaokrouhleno a vráceno.

```
def vytvorSeminko( predesli = 0, size=2**16):  
    predesli = predesli % size  
    if predesli == 0:  
        predesli = size  
  
    cpu_times = psutil.cpu_times()  
    seminko = cpu_times.user * cpu_times.idle % (predesli) + 1  
    seminko *= datetime.datetime.now().timestamp() * (10**6) % (10**6)  
    seminko *= cisloZeStringu(my_system.release)  
    seminko /= cisloZeStringu(my_system.processor)  
    seminko *= cisloZeStringu(my_system.node)  
    return round(seminko % size)
```

Zde je funkce vygenerujSeminka která vygeneruje daný počet semínek v určité velikosti podle parametrů. Vytvoří se potřebný počet semínek voláním funkce vytvorSeminko a pokud jednotlivá semínka již existují v seznamu vytvořených semínek tak je uloží do seznamu duplicit (stejneSeminka) pro analýzu.

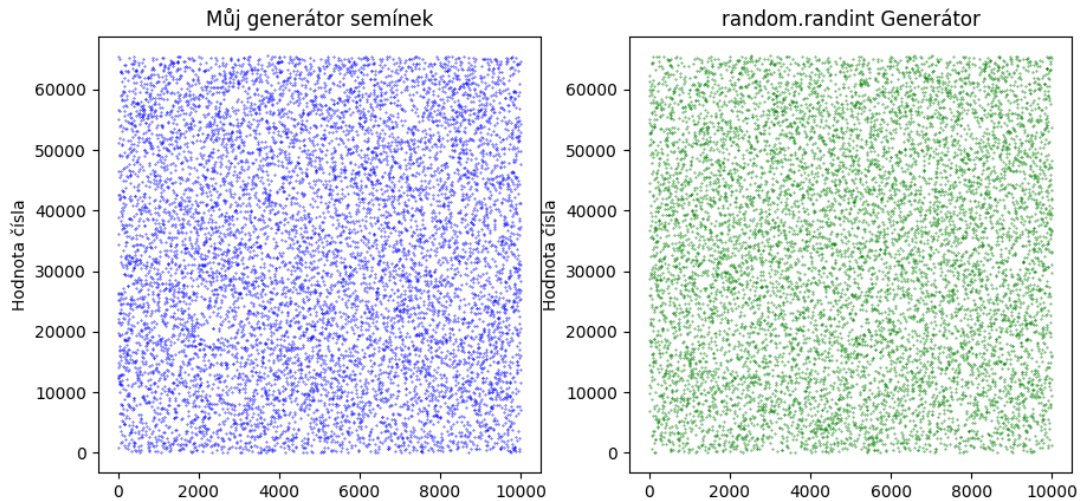
```
40 def vygenerujSeminka( pocet, size=2**16):
41     seminka = list()
42     stejneSeminka = list()
43     for i in range(pocet):
44         seminko = vytvorSeminko(sum(seminka),size)
45         if seminko in seminka:
46             stejneSeminka.append((i,seminko))
47         seminka.append(seminko)
48     return (seminka,stejneSeminka)
49
50 vystup = vygenerujSeminka(10000,2**32)
51
```

Zde mám jen pro porovnání vytvoření 10000 semínek za pomoci knihovny random. Následují jej výpisy průměrných hodnot vygenerovaných semínek, počet duplicitních čísel a čas běhu programu. Poslední část programu je vytvoření grafu pro zobrazení rozptylu.

```
55 testSeminekRandom = list()
56
57 for i in range(10000):
58     testSeminekRandom.append(random.randint(0,2**32))
59
60 print(numpy.average(vystup[0]))
61 print(numpy.average(testSeminekRandom))
62 print(f"Duplicitních čísel:{len(vystup[1])}")
63 print(f"Program běžel na: {time.perf_counter() - timer:0.04f} sekund")
64
65 fig, (ax1, ax2) = plt.subplots(ncols=2)
66 ax1.set_ylabel("Hodnota čísla")
67 ax1.set_title("Můj generátor semínek")
68
69 ax1.scatter(range(0,len(vystup[0])),vystup[0], s=0.1, color="blue")
70
71
72 ax2.set_ylabel("Hodnota čísla")
73 ax2.set_title("random.randint Generátor")
74 ax2.scatter(range(0,len(testSeminekRandom)),testSeminekRandom, s=0.1, color="green")
75 print(f"Program běžel na: {time.perf_counter() - timer:0.04f} sekund")
76
77 plt.show()
78
```

## Grafy

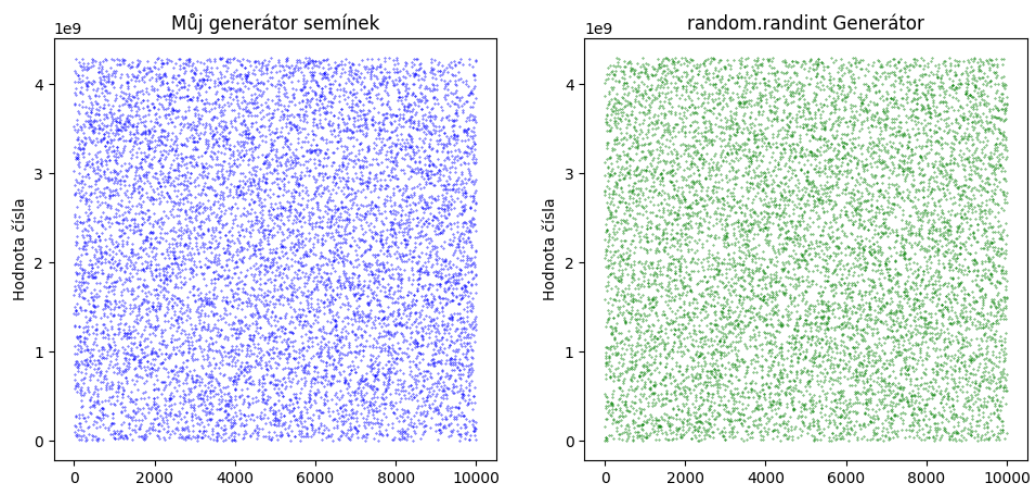
Vygenerované čísla jsem poté vložil do grafu, aby byl vidět rozptyl vygenerovaných čísel. Lze vidět, že rozptyl není špatný a že čísla jsou dostatečně rozprostřena.



Obrázek 1 Grafy pro 10000 vygenerovaných čísel s hodnotou mezi 0 a  $2^{16}$  (0 - 65 535)

```
Generátor běžel na: 1.6280 sekund  
32673.5065  
32968.313  
Duplicitních čísel:753
```

Obrázek 2 Výpis na konci programu, který obsahuje: délku běhu, průměrnou hodnotu semínek z mého generátoru, průměrnou, hodnotu semínek knihovny random, počet duplicitních čísel



Obrázek 3 Grafy pro 10000 vygenerovaných čísel s hodnotou mezi 0 a  $2^{32}$  (0 - 4 294 967 295)

```
Generátor běžel na: 3.0725 sekund
2158016325.356
2123539652.163
Duplicitních čísel:0
```

Obrázek 4 Výpis na konci programu, který obsahuje: délku běhu, průměrnou hodnotu semínek z mého generátoru, průměrnou, hodnotu semínek knihovny random, počet duplicitních čísel