Sprawozdanie z zadania:
"Generatory liczbowe"

18 grudnia 2024

Autorzy:

- Maria Małasiewicz (lsfr, zamiana bitów na system dziesiętny),
- Maciej Pestka (napisanie funkcji lcg, przetesowanie generatora dla różych parametrów i ziaren),
- Zuzanna Strauss (sprawdzenie długości generowanych ciągów losowych)

Zadanie 1:

Zaimplementować generatory liczbowe i przetestować je dla różnych parametrów

1 Wymagania podstawowe

1. zaimplementować przynajmniej jeden wybrany generator:
   (a) funkcję lcg(a,b,mod,seed), która generuje liczby pseudolosowe dla parametrów a, b, mod i ziarna seed,
   (b) funkcję lfsr(p,seed), która generuje pseudolosowy ciąg bitów dla współczynników p i ziarna seed,
   (c) inny generator,
2. przetestować generator dla różnych parametrów i ziaren,
3. sprawdzić długość generowanych ciągów losowych,

2 Wymagania dodatkowe

4. zaimplementować oba generatory,
5. bity wygenerowane w LFSR zamienić na liczby w systemie dziesiętnym,
6. dokonać próby wygenerowania najdłuższego pseudolosowego ciągu przy użyciu LCG oraz LFSR i porównać wyniki.

3 Lista spełnionych wymagań:

1, 2, 3, 4, 5, 6

4 Opis realizacji zadania:

4.1 Zaimplementowanie lfsr(p,seed), która generuje pseudolosowy ciąg bitów dla współczynników p i ziarna seed

Powstała funkcja `LSFR(p, seed)`, która na początku sprawdza, czy `seed` nie jest krótszy niż `p`. Jeśli $p > seed$ to funkcja zwraca błąd `IndexError("seed krótsze niż p!!")`.
Tworzona jest lista S zawierająca ostatnie `len(p)` elementów z `seed`. Następnie przy pomocy pętli obliczny jest nowy element `seed`. Iterujemy przez listę S i p obliczając $\sum_{i=1}^{n}(S_i \cdot p_i) \mod 2$,gdzie n jest długością listy p. Ostatecznie zwracana jest wydłużona o nowy element lista `seed`.

4.2 Przetestowanie generatora dla różnych parametrów i ziaren

Generator przetesowano na ziarne, który był wygenerowany na postawie czasu komputera. Seed został na podstawie jednego czasu komputeroewa wygeneorwał aż 10 różnych seedów. Powstał na postawie dzielenie lub mnożenia priewszego seeda, który został wygenerowany bez zmian.

4.3 sprawdzienie długości generowanych ciągów losowych

Stworzona jest funkcja `F(p, seed)`, która w pętli oblicza nowe `seed` jako `LFSR(p, seed)`. W każdej iteracji seed jest dzielony na dwie połowy, i jest sprawdzane, czy są one identyczne. Jeżeli tak, to zwracany jest ciąg bitów który generuje `seed` tzn.
dla ciągu bitów $B = [b_1, b_2, ..., b_n]$, ziarno $S = [b_1, b_2, .., b_n, b_1, b_2, ...]$.

4.4 zaimplementować oba generatory

Drugim generator był był `lcg(a,b,mod,seed)`. Który przyjmuje cztery argumenty, gdzie seed jest randomowym seedem, który był tworzony na postawie czasu komputera. A mod był wybierany losowo, przez testera. Oraz pozostałe parametry tak samo zostały wygenerowanie losowo.

4.5 bity wygenerowane w LFSR zamienić na liczby w systemie dziesiętnym,

W celu zamiany bitów wygenerowanych przez `LFSR` powstała funkcja `dziesietny(seed)`, gdzie `seed` może być listą kolejnych bitów, albo zapisem binarnym liczby podanym jako string.

Za pomocą pętli obliczmy $\sum_{i=0}^{n} s_i \cdot 2^{n-i}$, gdzie n jest długością `seed` a s_i jej i-tym elementem. Ostatecznie zwracana jest otrzymana suma, będąca wartością `seed` w systemie dziesiętnym.

4.6 dokonać próby wygenerowania najdłuższego pseudolosowego ciągu przy użyciu LCG oraz LFSR i porównać wyniki

Dla LFSR udało się wygeneorwać najdłuższy szyf dla `p=[[1,1,0]]` i dla `seed=[0,0,1]` ciąg wygeneorwać długość za pomocą funkcji której można podać jaki chcemy mieć maksymalny ciąg liczbowy. Podano argument, że ma ciąg liczbowy być długości 9999, który się udało wygenerować, natomiast dla dla 10000 konsolo odmówiła mi posłuszeństwa z wyświetleneniem tej liczby w systemiee dziesiętnym. Natomiast dla drugiej funkcji podawano duże wartości argumentów by powstała duży ciąg liczbowy. Ustawiono duży seed oraz by wartość mnożeszenie była największa oraz mod tak samo by była największa liczba.Próba w drugiej metodzie nie udało się wygenerować największa liczby, ponieważ wartości argumentów nie zmierzyła się na ekranie programu.

5 Prawa autorskie do kodu:

"Osidaczamy, iż praca została wytworzona samodzielne i bez wykorzystania narzędzi GenAl."

6 Wnioski:

Przy długich wartościach argumentów możmna się pomylić jakie dokładnie wartości były wpiszanie. zwłaszcza przy próbie wygenerowania największej liczby.

7 Źródła

Ślajdy ze laboratorium.