



od 1-25 losowy powielamy 3 krotnie i dopiero pootem :zadanie to ciąg losowy od 0-9 pozycji 75 i zapis tego ciągu najpierw pierwszych 25 pozycji takimi jakie sa pozniej od konca czyli 26 p[ozycji do 50 od konca a potem od 51 do 75 w dwouch partiacjh najpierw od srdka czyli 13 pozycji do poczatku i od 14 pozycji do konca czyli 25

&lt; 3 / 3 &gt;



Rozumiem, że chcesz **wygenerować ciąg losowy** o długości **75 pozycji** (cyfry od **0 do 9**) w **trzech krokach**:

1. **Pierwsze 25 pozycji**: Losowy ciąg od **0 do 9**.
2. **Pozycje 26-50**: Odwrócony ciąg z pozycji **1-25**.
3. **Pozycje 51-75**: Podzielony na **dwie części**:
  - **Pozycje 51-63**: Pierwsze **13 pozycji** od środka do początku.
  - **Pozycje 64-75**: Kolejne **12 pozycji** od środka do końca.

Postaram się to zrealizować krok po kroku.

## Krok 1: Generowanie pierwszych 25 pozycji

Generujemy **losowy ciąg** o długości **25 pozycji**, gdzie każda cyfra jest losowana z zakresu **0-9**.

### Przykład:

3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

## Krok 2: Generowanie pozycji 26-50 (odwrócony ciąg)

Odwracamy ciąg z **pozycji 1-25** i przypisujemy go do **pozycji 26-50**.

### Przykład:

Odwrócony ciąg:

9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3

## Krok 3: Generowanie pozycji 51-75

Dzielimy ciąg z **pozycji 1-25** na **dwie części**:

1. **Pozycje 51-63**: Pierwsze **13 pozycji** od środka do początku.
2. **Pozycje 64-75**: Kolejne **12 pozycji** od środka do końca.

### Środek ciągu:

Dla ciągu o długości **25**, środek to **13. pozycja**.

### Przykład:

- **Pozycje 51-63**:  
3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1
- **Pozycje 64-75**:  
4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

## Pełny ciąg 75-pozycyjny

Łączymy wszystkie trzy części:

**1. Pozycje 1-25:**

3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

**2. Pozycje 26-50:**

9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3

**3. Pozycje 51-75:**

3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

**Pełny ciąg:**

3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

## Implementacja w Pythonie

Oto przykładowy kod w Pythonie, który realizuje to zadanie:

```
python
```

Copy

```
import random
```

```
# Krok 1: Generowanie pierwszych 25 pozycji
```

```
first_25 = [random.randint(0, 9) for _ in range(25)]
```

```
# Krok 2: Generowanie pozycji 26-50 (odwrócony ciąg)
```

```
reversed_25 = first_25[::-1]
```

```
# Krok 3: Generowanie pozycji 51-75
```

```
middle = 13 # Środek ciągu
```

```
positions_51_63 = first_25[:middle] # Pierwsze 13 pozycji
```

```
positions_64_75 = first_25[middle:] # Kolejne 12 pozycji
```

```
# Łączenie wszystkich części
```

```
full_sequence = first_25 + reversed_25 + positions_51_63 + positions_64_75
```

```
# Wyświetlenie wyniku
```

```
print("Pełny ciąg 75-pozycyjny:")
```

```
print(full_sequence)
```

## Wynik działania kodu

Pełny ciąg 75-pozycyjny będzie miał postać:

Copy

```
[3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9]
```

## Podsumowanie

**1. Pierwsze 25 pozycji:** Losowy ciąg od 0 do 9.

2. **Pozycje 26-50:** Odwrócony ciąg z pozycji **1-25**.

3. **Pozycje 51-75:** Podzielony na **dwie części** (od środka do początku i od środka do końca).

Dzięki temu podejściu otrzymujemy **spójny i powtarzalny** ciąg 75-pozycyjny, który może być wykorzystany w różnych zastosowaniach (np. algorytmy, symulacje).

czyli tak czy owak mamy do czynienia z unikalną pozycją ponieważ ciąg (25 pozycji) zawsze może być różny a kolejne dokładki z niego wynikające także unikalne tworzą adresację



Tak, dokładnie! **Unikalność ciągu 25-pozycyjnego** oraz **jego pochodnych** (odwrócony ciąg i podciągi od środka) sprawia, że cała struktura **75-pozycyjna** jest **unikalna** i może służyć jako **adresacja** dla konkretnego układu danych (np. obrazu, sekwencji, itp.). Oto dlaczego:

## 1. Unikalność ciągu 25-pozycyjnego

- Każda z **25 pozycji** jest losowana niezależnie z zakresu **0-9**.
- Liczba możliwych kombinacji:  $10^{25}$  (czyli **10 sekstylionów**).
- Nawet jeśli powielamy ten ciąg w różnych formach (odwrócony, podciągi), to **pierwotny ciąg** pozostaje **unikalny**.

## 2. Unikalność pochodnych ciągów

- **Odwrócony ciąg (pozycje 26-50):**
  - Jest **bezpośrednio zależny** od pierwotnego ciągu.
  - Jeśli pierwotny ciąg jest unikalny, to jego odwrócenie również jest unikalne.
- **Podciągi od środka (pozycje 51-75):**
  - Są **wycinkami** pierwotnego ciągu, ale ich **kolejność i zawartość** zależą od **struktury pierwotnego ciągu**.
  - Nawet jeśli podciągi mogą się powtarzać w różnych ciągach, to ich **połączenie z pierwotnym ciągiem** tworzy **unikalną całość**.

## 3. Adresacja 75-pozycyjna

- **Pierwsze 25 pozycji:** Unikalny ciąg losowy.
- **Pozycje 26-50:** Odwrócony ciąg, który jest **unikalny** dla danego ciągu pierwotnego.
- **Pozycje 51-75:** Podciągi od środka, które są **unikalne** w kontekście całej struktury.

### Przykład:

Dla ciągu pierwotnego:

3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

- **Odwrócony ciąg:**  
9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3, 6, 0, 8, 5, 2, 9, 4, 1, 7, 3
- **Podciągi od środka:**
  - Pierwsze 13 pozycji: 3, 7, 1, 4, 9, 2, 5, 8, 0, 6, 3, 7, 1
  - Kolejne 12 pozycji: 4, 9, 2, 5, 8, 0, 6, 3, 7, 1, 4, 9

**Pełny ciąg 75-pozycyjny jest unikalny**, ponieważ:

- Nawet jeśli **podciągi** mogą się powtarzać w innych ciągach, to ich **kombinacja z pierwotnym ciągiem i odwróconym ciągiem** tworzy **niepowtarzalną strukturę**.

## 4. Zastosowanie jako adresacja

Ta struktura może być użyta jako **adresacja** dla:

### 1. Obrazów:

- Każdy obraz może być reprezentowany przez **unikalny ciąg 75-pozycyjny**.
- Adresacja pozwala na **szybkie wyszukiwanie i porównywanie** obrazów.

### 2. Sekwencji danych:

- Np. sygnały, dźwięki, sekwencje genetyczne.
- Unikalny ciąg może służyć jako **identyfikator** dla konkretnej sekwencji.

### 3. Baz danych:

- Każdy rekord może mieć **unikalny adres** w postaci ciągu 75-pozycyjnego.
- Uproszcza to **indeksowanie i wyszukiwanie**.

## 5. Przykład praktyczny

**Adresacja obrazu:**

### 1. Generowanie ciągu:

- Obraz jest przetwarzany na **ciąg 25-pozycyjny** (np. na podstawie kolorów pikseli).
- Następnie generujemy **odwrócony ciąg i podciągi od środka**.

### 2. Zapis adresu:

- Pełny ciąg 75-pozycyjny jest zapisywany jako **unikalny adres** obrazu.

### 3. Wyszukiwanie:

- Aby znaleźć obraz, wystarczy **porównać jego adres** z adresami w bazie danych.

## Podsumowanie

- **Ciąg 25-pozycyjny** jest **unikalny** dzięki losowości i ogromnej liczbie kombinacji ( $10^{25}$ ).
- **Pochodne ciągi** (odwrócony i podciągi) są **unikalne** w kontekście pierwotnego ciągu.
- **Pełny ciąg 75-pozycyjny** może służyć jako **adresacja** dla różnych rodzajów danych (obrazy, sekwencje, rekordy).

Dzięki temu podejściu mamy **skalowalny i unikalny system adresacji**, który może być wykorzystany w wielu zastosowaniach.

okazalo sie to prawda ze tym sposobem z 25 do 75 pozycji da sie brnac dalej