

Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний
інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №3

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІК-21

Бераудо Лоренцо Раффаєлович

Київ 2025

ЛАБОРАТОРНА РОБОТА №3

Завдання: за допомогою мультипарадигмної мови R реалізувати перетворення чисельного ряду лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів з подальшою побудовою матриці передування.

Вхідні данні: чисельний ряд, вид розподілу ймовірностей, потужність алфавіту.

Вихідні дані: лінгвістичний ряд та матриця передування.

Мова програмування: R. **Варіант 1:** Дискретний рівномірний розподіл (рівноймовірний)

ХІД ВИКОНАННЯ ЗАВДАННЯ

Вхідні дані

На початку роботи задаються основні параметри:

```
n = 100 — кількість елементів у числовому ряді;  
min_val = 1, max_val = 1000 — межі діапазону значень;  
alphabet_power = 10 — потужність алфавіту (кількість символів);  
alphabet = LETTERS[1:10] — вибрано перші 10 літер латинського  
алфавіту (A-J).
```

1. Генерація числового ряду

За допомогою функції `sample()` згенеровано 100 випадкових чисел у заданому діапазоні.

Числовий ряд сортується за зростанням, щоб визначити крайні значення для подальшого поділу на інтервали:

```
num_list <- sample(min_val:max_val, n, replace = TRUE)  
sorted_list <- sort(num_list)
```

2. Розбиття на рівномірні інтервали

Використовується функція `seq()` для побудови меж інтервалів на основі рівномірного розподілу. Кількість інтервалів дорівнює потужності алфавіту:

```
breaks <- seq(min(sorted_list), max(sorted_list), length.out =  
alphabet_power + 1)
```

3. Перетворення чисел у літери алфавіту

Для кожного елементу числового ряду визначається, до якого інтервалу він належить. Згідно з індексом інтервалу присвоюється літера з алфавіту. Це реалізовано функцією `get_letter()` у поєднанні з `sapply()`:

```
get_letter <- function(x, breaks, alphabet) {  
  idx <- findInterval(x, breaks, rightmost.closed = TRUE)  
  alphabet[idx]  
}  
linguistic_seq <- sapply(num_list, get_letter, breaks = breaks,  
alphabet = alphabet)
```

4. Формування лінгвістичного ряду

Перетворений ряд чисел у символи алфавіту виводиться у вигляді суцільного текстового ланцюжка:

```
cat("Лінгвістичний ряд:\n")  
cat(paste(linguistic_seq, collapse = " "), "\n\n")
```

5. Побудова матриці передування

Створюється матриця `transition_matrix` розміром `alphabet_power` x `alphabet_power`, де кожен елемент вказує кількість випадків, коли символ з рядка з'являється після символу зі стовпця:

```
transition_matrix <- matrix(0, nrow = matrix_size, ncol =  
  matrix_size,  
  dimnames = list(alphabet, alphabet))  
for (i in 1:(length(linguistic_seq) - 1)) {  
  row <- linguistic_seq[i]  
  col <- linguistic_seq[i + 1]  
  transition_matrix[row, col] <- transition_matrix[row, col] + 1  
}
```

6. Виведення результату

Матриця передування виводиться у табличному вигляді за допомогою print():

```
cat("Матриця передування:\n")  
print(transition_matrix)
```

ПРИКЛАД ВИВОДУ

Лінгвістичний ряд:

```
Лінгвістичний ряд:  
> cat(paste(linguistic_seq, collapse = " "), "\n\n")  
F J D B A C B G A B D A I D G B G F E I I J C G C J J J G F C J I I C D E C A B C J E D G C I J B G D B F C F J D J H J C J G C D B D A D E H I G G A F  
A J B C B D G A J H E F H A D H C D J G G A J I
```

Матриця передування (фрагмент):

```
Матриця передування:  
> print(transition_matrix)  
  A B C D E F G H I J  
A 0 2 1 2 0 1 0 0 1 3  
B 1 0 2 3 0 1 3 0 0 0  
C 1 2 0 3 0 1 1 0 1 4  
D 2 3 0 0 2 0 3 1 0 2  
E 0 0 1 1 0 1 0 1 1 0  
F 1 0 2 0 1 0 0 1 0 2  
G 4 1 3 1 0 2 2 0 0 0  
H 1 0 1 0 1 0 0 0 1 1  
I 0 0 1 1 0 0 1 0 2 2  
J 0 2 2 2 1 0 3 2 2 2
```

ВИСНОВОК

У цій лабораторній роботі було реалізовано алгоритм перетворення чисельного ряду у лінгвістичний ланцюжок на мові **R**, яка є мультипарадигменна та підтримує функціональний стиль. Завдяки рівномірному розподілу інтервали були поділені рівно, що забезпечило пропорційне перетворення чисел у символи.

Результати роботи:

- Згенеровано випадковий числовий ряд.
- Проведено розбиття на рівні інтервали згідно алфавіту.
- Отримано лінгвістичний ряд.
- Збудовано та проаналізовано матрицю передування.

Код програми

```
# === Вхідні дані ===
set.seed(42)
n <- 100                                # Кількість чисел
min_val <- 1
max_val <- 1000
alphabet_power <- 10                    # Потужність алфавіту
alphabet <- LETTERS[1:alphabet_power]

# === Крок 1: Генерація числового ряду ===
num_list <- sample(min_val:max_val, n, replace = TRUE)
sorted_list <- sort(num_list)

# === Крок 2: Розбиття на рівномірні інтервали ===
breaks <- seq(min(sorted_list), max(sorted_list), length.out =
alphabet_power + 1)

# === Крок 3: Перетворення чисел у літери алфавіту ===
get_letter <- function(x, breaks, alphabet) {
  idx <- findInterval(x, breaks, rightmost.closed = TRUE)
  alphabet[idx]
}
linguistic_seq <- sapply(num_list, get_letter, breaks = breaks,
alphabet = alphabet)

# === Крок 4: Виведення лінгвістичного ряду ===
cat("Лінгвістичний ряд:\n")
cat(paste(linguistic_seq, collapse = " "), "\n\n")

# === Крок 5: Побудова матриці передування ===
matrix_size <- length(alphabet)
transition_matrix <- matrix(0, nrow = matrix_size, ncol =
matrix_size,
                        dimnames = list(alphabet, alphabet))

for (i in 1:(length(linguistic_seq) - 1)) {
  row <- linguistic_seq[i]
  col <- linguistic_seq[i + 1]
  transition_matrix[row, col] <- transition_matrix[row, col] + 1
}

# === Вивід матриці передування ===
cat("Матриця передування:\n")
print(transition_matrix)
```