



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №1
Мультипарадигменне програмування

Виконав
студент групи ІА-23:
Курач В.А.

Київ 2025

Завдання: на процедурній мові програмування реалізувати перетворення чисельного ряду до 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: Фортран

Хід розв'язання задачі:

1. Чисельний ряд сортується від найменшого значення до найбільшого. Тим самим отримуємо діапазон (область) припустимих значень.
2. ОПЗ розбиваємо на інтервали (кількість залежить від потужності обраного алфавіту) у відповідності до розподілу ймовірностей, при цьому враховуємо, що ймовірність потрапляння до інтервалу $P[a,b]=F(b)-F(a)$.
3. Кожному числовому значенню ставиться у відповідність знак з алфавіту, який має той самий індекс, що і інтервал.
4. Після заміни отриманий ряд літер видається на вихід.
5. На основі лінгвістичного ряду будується матриця

	A	B	...	Z
A	$a_{1,1}$	$a_{1,2}$...	$a_{1,26}$
B	$a_{2,1}$	$a_{2,2}$...	$a_{2,26}$
...
Z	$a_{26,1}$	$a_{26,2}$...	$a_{26,26}$

де $a_{1,2}$ — кількість в лінгвістичному ряду випадків літери B після літери A.

Варіанти.

№ варіанту	Розподіл ймовірностей
1	Дискретний рівномірний розподіл (рівноймовірний)

Хід роботи

- Програма читає друге поле з кожного рядка CSV-файлу.
- Значення зберігаються у масиві `series`.
- Визначається мінімальне та максимальне значення ряду, після чого діапазон ділиться на N інтервалів однакової довжини.
- Кожне число відноситься до певного інтервалу та замінюється символом алфавіту (A, B, ..., Z).
- Побудований лінгвістичний ряд виводиться на екран.
- Далі формується **матриця передування** — кількість випадків, коли за символом X слідує символ Y.

Обрали B-C-D-E-F-Brent Oil Futures Historical Data.csv


```

end if

! 2. Введення алфавіту
print *, "Введіть потужність алфавіту (не більше", max_alpha, "):"
read *, alpha_size
if (alpha_size > max_alpha .or. alpha_size < 1) stop "Невірна потужність алфавіту"

do i = 1, alpha_size
    alphabet(i) = char(64 + i)
end do

! 3. Копіюємо масив для сортування (щоб зберегти оригінал)
series_copy(1:n) = series(1:n)
call sort_array(series_copy, n)
min_val = series_copy(1)
max_val = series_copy(n)
interval = (max_val - min_val) / alpha_size

! 4. Перетворення в лінгвістичний ряд (по оригінальному порядку)
do i = 1, n
    linguistic_series(i) = alphabet(alpha_size) ! за замовчуванням остання літера
    do j = 1, alpha_size
        if (series(i) <= min_val + j * interval) then
            linguistic_series(i) = alphabet(j)
            exit
        end if
    end do
end do

! 5. Вивід лінгвістичного ряду
print *, "Лінгвістичний ряд:"
do i = 1, n
    write(*, "(A1)", advance='no') linguistic_series(i)
end do
print *

! 6. Матриця передування
transition_matrix = 0
do i = 1, n - 1
    current = index_in_alphabet(linguistic_series(i), alphabet, alpha_size)
    next = index_in_alphabet(linguistic_series(i+1), alphabet, alpha_size)
    if (current > 0 .and. next > 0) then
        transition_matrix(current, next) = transition_matrix(current, next) + 1
    end if
end do

! 7. Вивід матриці передування
print *, "Матриця передування:"
write(*, '(A3)', advance='no') " "
do i = 1, alpha_size
    write(*, '(A3)', advance='no') alphabet(i)
end do
print *

do i = 1, alpha_size
    write(*, '(A3)', advance='no') alphabet(i)
    do j = 1, alpha_size
        write(*, '(I3)', advance='no') transition_matrix(i,j)
    end do
end do

```

```
    print *  
end do
```

contains

```
subroutine sort_array(arr, n)  
  real, intent(inout) :: arr(:)  
  integer, intent(in) :: n  
  integer :: i, j  
  real :: temp
```

```
  do i = 1, n-1  
    do j = i+1, n  
      if (arr(i) > arr(j)) then  
        temp = arr(i)  
        arr(i) = arr(j)  
        arr(j) = temp  
      end if  
    end do  
  end do
```

```
end subroutine sort_array
```

```
integer function index_in_alphabet(symbol, alphabet, size)  
  character(len=1), intent(in) :: symbol  
  character(len=1), dimension(size), intent(in) :: alphabet  
  integer, intent(in) :: size  
  integer :: k
```

```
  index_in_alphabet = -1  
  do k = 1, size  
    if (alphabet(k) == symbol) then  
      index_in_alphabet = k  
      return  
    end if  
  end do
```

```
end function index_in_alphabet
```

```
subroutine extract_field(line, field_num, result)  
  character(len=*), intent(in) :: line  
  integer, intent(in) :: field_num  
  character(len=*), intent(out) :: result  
  integer :: i, count, start, endp
```

```
  count = 0  
  start = 1  
  endp = len_trim(line)
```

```
  do i = 1, len_trim(line)  
    if (line(i:i) == ',') then  
      count = count + 1  
      if (count == field_num - 1) start = i + 1  
      if (count == field_num) then  
        endp = i - 1  
        exit  
      end if  
    end if  
  end do
```

```
  if (count < field_num - 1) then
```

```
        result = "  
    else  
        result = adjustl(line(start:endp))  
        if (len_trim(result) > 0) then  
            if (result(1:1) == "") result = result(2:)  
            do i = len_trim(result), 1, -1  
                if (result(i:i) == "") then  
                    result(i:) = '  
                    exit  
                end if  
            end do  
            result = adjustl(result)  
        end if  
    end if  
end subroutine extract_field  
  
end program linguistic_chain
```

Висновки: У результаті виконання лабораторної роботи було успішно реалізовано перетворення числового ряду в лінгвістичний. Побудовано матрицю передування, яка може бути використана для подальшого аналізу залежностей між символами. Отримано практичні навички роботи з масивами, сортуванням, обробкою CSV та реалізацією власних процедур у Fortran.