



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

Мультипарадигмене програмування

Виконав:
Студент групи ІА-12
Бутрій В.А.

Київ 2024

Мета роботи: на імперативній мові програмування реалізувати перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів з подальшою побудовою матриці передування.

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.

Хід роботи

Згідно завдання, було реалізовано програму процедурною мовою програмування (в моєму випадку це C).

Код проекту можна побачити тут: <https://github.com/vetterocki/MPP/tree/main/lab1>

Програма підтримує зміну потужності алфавіту, зміну самого алфавіту (наприклад, можемо вилючити певні літери з послідовності) та, очевидно, чисельного ряду. Було обрано варіант, коли розподіл ймовірностей в нас працює як розбиття на рівні інтервали. Програма було логічно розподілена на функції, що послідовно викликаються в main():

```
#define ALPHABET_POWER 3
#define ARRAY_SIZE 6

void setConsoleOutputUTF8();

void sortArray(const int array[], int sortedArray[]);

void createIntervalMap(const int sortedArray[], char intervalMap[]);

void mapValuesToChars(const int array[], const char intervalMap[], char chars[]);

void generateMatrix(char chars[], char matrix[][ALPHABET_POWER]);

int getCount(const char *chars, char from, char to);

char alphabet[] = { [0]: 'A', [1]: 'B', [2]: 'C', [3]: 'D', [4]: 'E', [5]: 'F', [6]: 'G',
                    [7]: 'H', [8]: 'I', [9]: 'J', [10]: 'K', [11]: 'L', [12]: 'M', [13]: 'N',
                    [14]: 'O', [15]: 'P', [16]: 'Q', [17]: 'R', [18]: 'S', [19]: 'T',
                    [20]: 'U', [21]: 'V', [22]: 'W', [23]: 'X', [24]: 'Y', [25]: 'Z'};

int main() {
    setConsoleOutputUTF8();

    int array[ARRAY_SIZE] = { [0]: 4, [1]: 1, [2]: 6, [3]: 3, [4]: 2, [5]: 5};
    int sortedArray[ARRAY_SIZE];
    char intervalMap[ARRAY_SIZE];
    char chars[ARRAY_SIZE];
    char matrix[ALPHABET_POWER][ALPHABET_POWER];

    sortArray(array, sortedArray);
    createIntervalMap(sortedArray, intervalMap);
    mapValuesToChars(array, intervalMap, chars);
    generateMatrix(chars, matrix);
}
```

Наведу приклад роботи програми. Нехай дано чисельний ряд [9, 1, 6, 7, 3, 4, 5, 10] та потужність алфавіту 4. Буду використовувати повний алфавіт, отже, символи, які будуть використані, це A, B, C, D.

Інтервали: [1, 3] - A; [4, 5] - B; [6, 7] - C; [9, 10] - D.

Результуючий лінгвістичний ряд: D A C C A B B D

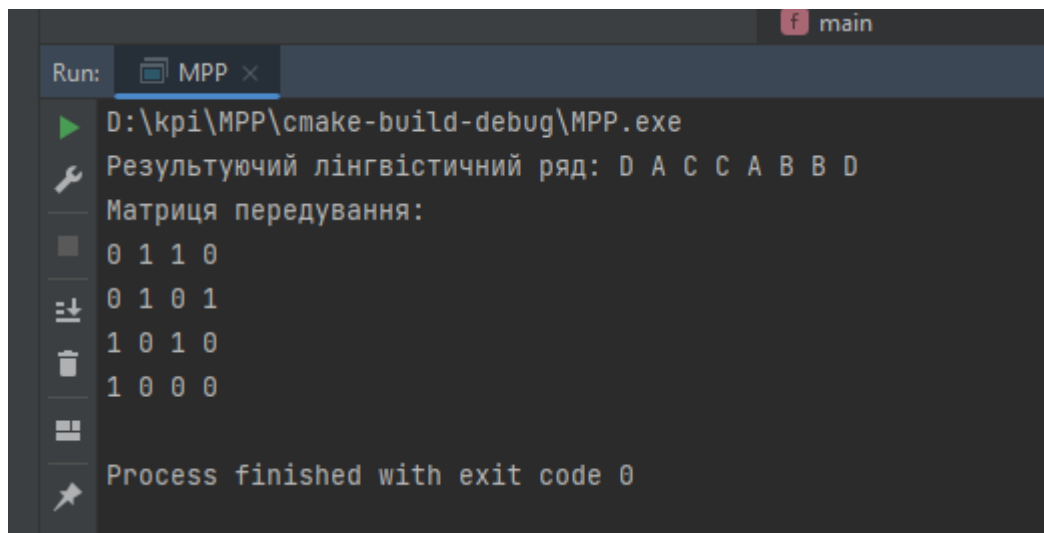
Результуюча матриця передування:

	A	B	C	D
A	0	1	1	0
B	0	1	0	1
C	1	0	1	0
D	1	0	0	0

Перевіримо, які результати порахує програма:

```
#define ALPHABET_POWER 4  
#define ARRAY_SIZE 8
```

```
int array[ARRAY_SIZE] = { [0]: 9, [1]: 1, [2]: 6, [3]: 7, [4]: 3, [5]: 4, [6]: 5, [7]: 10};
```



```
Run: MPP x  
D:\kpi\MPP\sake-build-debug\MPP.exe  
Результуючий лінгвістичний ряд: D A C C A B B D  
Матриця передування:  
0 1 1 0  
0 1 0 1  
1 0 1 0  
1 0 0 0  
Process finished with exit code 0
```

Як можемо побачити, результати мануального підрахунку збіглися с програмними результатами.

Висновок:

Під час виконання лабораторної роботи я на імперативній мові програмування (а конкретно на мові C) реалізував перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів з подальшою побудовою матриці передування.