

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Катедра «Комп'ютерна інженерія та програмування»

ЗВІТ

про виконання лабораторної роботи №2
з навчальної дисципліни «Алгоритми та структури даних»

Варіант 9

Виконав студент:

Ульянов Кирило Юрійович

Група: КН-1023b

Перевірив:

старший викладач

Бульба С.С.

Харків-2024

1 Мета роботи

Набути навичок та практичного досвіду у розробці рекурсивних програм.

2 Хід роботи

1) Розробити рекурсивний та ітераційний алгоритми розв'язання індивідуального завдання. Визначити та порівняти час виконання відповідних функцій, зробити висновки:

9. Обчислити корінь рівняння $f(x) = x^3 - 2x^2 - 3x + 10$ із заданою точністю в інтервалі $a \leq x \leq b$ методом дихотомії.

2) Створюю базовий метод для обчислення значень моєї функції:

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <time.h>
5
6  #include "general_utils.h"
7
8  double f(double x) { return x * x * x - 2 * x * x - 3 * x + 10; }
```

3) Створюю ітераційний алгоритм методу дихотомії:

```

1      double iterationBisection(double a, double b, double epsilon) {
2          double mid = 0;
3
4          if (f(a) * f(b) >= 0) {
5              printf("The function has no root on that interval [%f, %f]\n", a
6                  , b);
7              exit(0);
8          }
9
10         while ((b - a) >= epsilon) {
11             mid = (a + b) / 2;
12
13             if (fabs(f(mid)) <= epsilon) {
14                 break;
15             }
16
17             if (f(mid) * f(a) < 0) {
18                 b = mid;
19             } else {
20                 a = mid;
21             }
22         }
23
24         return mid;
25     }
```

4) Створюю **рекурсивний** алгоритм методу дихотомії:

```

1  double recursiveBisection(double a, double b, double epsilon) {
2      double mid = (a + b) / 2;
3
4      if (fabs(f(mid)) <= epsilon) {
5          return mid;
6      }
7
8      if (f(a) * f(b) >= 0) {
9          printf("The function has no root on that interval [%f, %f]\n", a
10              , b);
11          exit(0);
12      }
13
14      if ((b - a) < epsilon) {
15          return mid;
16      } else if (f(mid) * f(a) < 0) {
17          return recursiveBisection(a, mid, epsilon);
18      } else {
19          return recursiveBisection(mid, b, epsilon);
20      }
21 }

```

5) Створюю функцію яка буде тестувати **ітераційний** алгоритм:

```

1  void iterationBisectionTest(double a, double b, double epsilon, int
2      test_iterations) {
3      double f_root = 0;
4      double speed_time = 0, avg_speed_time = 0, avg_test = 0;
5      clock_t start, end;
6
7      for (size_t i = 0; i < test_iterations; i++) {
8          start = clock();
9          f_root = iterationBisection(a, b, epsilon);
10         printf("The root of the equation on the interval \033[33m[%.2f, "
11             "%.2f]\033[0m: %.2f\n",
12             a, b, f_root);
13         end = clock();
14         speed_time = ((double)(end - start)) / CLOCKS_PER_SEC * 1000;
15         printf("Function execution time: \033[32m%.3f\033[0m ms.\n\n",
16             speed_time);
17         avg_speed_time += speed_time;
18     }
19
20     avg_test = avg_speed_time / test_iterations;
21
22     printf("AVERAGE EXECUTION TIME USING \033[34m%d\033[0m CALLS OF"
23         "\033[34m ITERATIVE \033[0m"
24         "METHOD IS: "
25         "\033[32m%.3f\033[0m milliseconds.\n\n\n",
26         test_iterations, avg_test);
27 }

```

6) Створюю функцію яка буде тестувати **рекурсивний** алгоритм:

```

1 void recursiveBisectionTest(double a, double b, double epsilon, int
  test_iterations) {
2     double f_root = 0;
3     double speed_time = 0, avg_speed_time = 0, avg_test = 0;
4     clock_t start, end;
5
6     for (size_t i = 0; i < test_iterations; i++) {
7         start = clock();
8         f_root = recursiveBisection(a, b, epsilon);
9         printf("The root of the equation on the interval \033[33m[%.2f, "
10              "%.2f]\033[0m: %.2f\n",
11              a, b, f_root);
12         end = clock();
13         speed_time = ((double)(end - start)) / CLOCKS_PER_SEC * 1000;
14         printf("Function execution time: \033[32m%.3f\033[0m ms.\n\n",
15              speed_time);
16         avg_speed_time += speed_time;
17     }
18
19     avg_test = avg_speed_time / test_iterations;
20
21     printf("AVERAGE EXECUTION TIME USING \033[34m%d\033[0m CALLS OF"
22           "\033[34m ITERATIVE \033[0m"
23           "METHOD IS: "
24           "\033[32m%.3f\033[0m milliseconds.\n\n\n",
25           test_iterations, avg_test);
26 }

```

7) Запускаю у головній функції виконання тестуючих методів, які в свою чергу, запускають обидва види алгоритму декілька разів та ми отримаємо:

- по декілька запусків кожного з алгоритмів з однаковими параметрами.
- час виконання кожного запуску алгоритму.
- середній час виконання порції алгоритмів для більш детальних результатів.

```

1 void task1() {
2     double a = -70, b = 70, epsilon = 0.0001, test_iterations = 5;
3
4     highlightText("Iterative bisection algorythm:", "blue");
5     iterationBisectionTest(a, b, epsilon, test_iterations);
6
7     highlightText("Recursive bisection algorythm:", "blue");
8     recursiveBisectionTest(a, b, epsilon, test_iterations);
9 }

```

7) Результати з параметрами $a = -5$, $b = 5$, $\epsilon = 0.0001$ на 5 повторів:

```

~/Doc/university/Al/l/build ./main
Доступні опції:
-1 -- вийти з програми
 1 -- Завдання 1

Введіть номер опціїї або номер завдання: 1

=====
Завдання 1
=====
Iterative bisection algorythm:
The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.011 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.005 ms.

AVERAGE EXECUTION TIME USING 5 CALLS OF ITERATIVE METHOD IS: 0.006 milliseconds.

```

```

Recursive bisection algorythm:
The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.006 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-5.00, 5.00]: -2.00
Function execution time: 0.004 ms.

AVERAGE EXECUTION TIME USING 5 CALLS OF ITERATIVE METHOD IS: 0.005 milliseconds.

```

8) Результати з параметрами $a = -20$, $b = 20$, $\epsilon = 0.0001$ на 5 повторів:

```
Iterative bisection algorithm:
The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.007 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.004 ms.

AVERAGE EXECUTION TIME USING 5 CALLS OF ITERATIVE METHOD IS: 0.005 milliseconds.
```

```
Recursive bisection algorithm:
The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-20.00, 20.00]: -2.00
Function execution time: 0.005 ms.

AVERAGE EXECUTION TIME USING 5 CALLS OF ITERATIVE METHOD IS: 0.005 milliseconds.
```


9) Результати з параметрами $a = -70$, $b = 70$, $\epsilon = 0.0001$ на 10 повторів:

```
Iterative bisection algorithm:
The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.008 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.017 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.
in ((a = -20), (b = 20), ((epsilon = 0.0001)) на 5 повторів:
The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.
ports/algos/lab2/assets/4.png
The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.
ports/algos/lab2/assets/5.png
The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.004 ms.

AVERAGE EXECUTION TIME USING 10 CALLS OF ITERATIVE METHOD IS: 0.006 milliseconds.
```

```
Recursive bisection algorithm:
The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.006 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

The root of the equation on the interval [-70.00, 70.00]: -2.00
Function execution time: 0.005 ms.

AVERAGE EXECUTION TIME USING 10 CALLS OF ITERATIVE METHOD IS: 0.005 milliseconds.
```

10) Також проведено декілька тестів з різною кількістю викликів алгоритмів:

```
Iterative bisection algorythm:  
AVERAGE EXECUTION TIME USING 50 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Recursive bisection algorythm:  
AVERAGE EXECUTION TIME USING 50 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Iterative bisection algorythm:  
AVERAGE EXECUTION TIME USING 100 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Recursive bisection algorythm:  
AVERAGE EXECUTION TIME USING 100 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Iterative bisection algorythm:  
AVERAGE EXECUTION TIME USING 1000 CALLS OF ITERATIVE METHOD IS: 0.001 milliseconds.
```

```
Recursive bisection algorythm:  
AVERAGE EXECUTION TIME USING 1000 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Iterative bisection algorythm:  
AVERAGE EXECUTION TIME USING 3000 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```

```
Recursive bisection algorythm:  
AVERAGE EXECUTION TIME USING 3000 CALLS OF ITERATIVE METHOD IS: 0.002 milliseconds.
```


3 Висновки

В ході виконання лабораторної роботи я реалізував метод дихотомії у рекурсивному та ітераційному виді. Можу зазначити що ітераційний та рекурсивний метод майже не мають сильної різниці коли відпрацьовують багато разів, результати виходять схожі. Але в окремих одиничних випадках все ж таки рекурсивний метод виявляється трішки швидкішим, але повільнішим при великому виклику рекурсивних процедур.