|  |  |
| --- | --- |
| |  | | --- | | МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |

Институт Информационных технологий

Кафедра МОСИТ

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3**

по дисциплине «Теория принятия решений»

**Студент группы** ИКБО-11-17 Алиев Ю. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента)*

**Руководитель работы** Петренко А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись руководителя)*

Москва, 2020

# Постановка задачи

Найти: *Argmin* *f*(*x*), *x* ∈ [-5; 5] при Δ*x* ≤ 0.1 перечисленными ниже методами в соответствии с вариантом.

1. Классический метод
2. Метод перебора
3. Метод поразрядного поиска

Вариант 1: *f*(*x*) = 2x2 - 5

# Классический метод

Найдем производную от квадратичной функции и приравняв ее к нули, посмотрим, где функция меняет знакопостоянство. Так как функция – квадратичная, проверки на концах не имеют смысла.

*f’*(*x*) = 4x = 0 из интервала *x* ∈ [-5; 5], где *x*1 = 0 (-5)=> min = *x*1

# Метод перебора

Число отрезков разбиения будет равно 5+5 /0,1 = 100. Тогда число отрезков n = 100.

Вычислим значения *f*(*xi*), где *xi* = -5 + 0.1*i*, *i* = 0, …, 100.

Для этого мной была написана программа на языке программирования JavaScript.

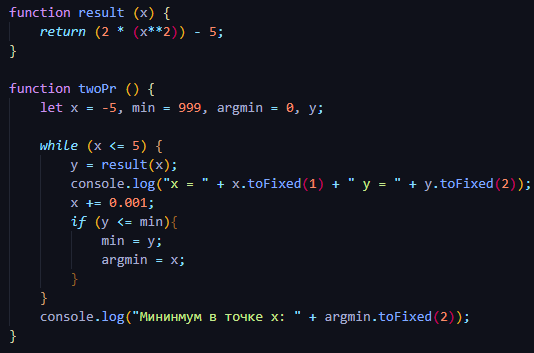


Рисунок 1. Код программы методом перебора

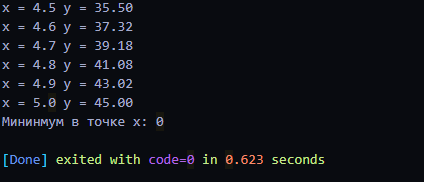


Рисунок 2. Результат программы при использовании метода перебора

# Метод поразрядного поиска

Начальный шаг Δ*x*1 = 0.3.

Для вычисления значения была написана следующая программа на языке программирования JavaScript, реализующая решение методом поразрядного поиска.

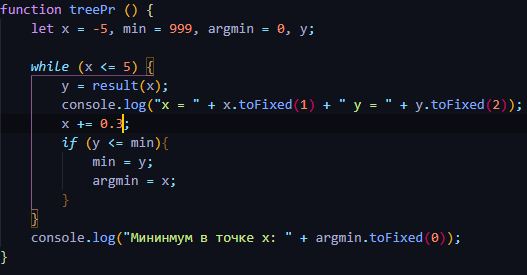


Рисунок 3. Реализация метода поразрядного поиска на языке Java

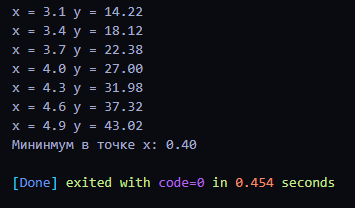


Рисунок 4. Результат программы при использовании метода поразрядного поиска и сравнение со вторым способом

Исходный код программы:

|  |
| --- |
| function result (x) {  return (2 \* (x\*\*2)) - 5;  }  function twoPr () {  let x = -5, min = 999, argmin = 0, y;  while (x <= 5) {  y = result(x);  console.log("x = " + x.toFixed(1) + " y = " + y.toFixed(2));  x += 0.001;  if (y <= min){  min = y;  argmin = x;  }  }  console.log("Мининмум в точке x: " + argmin.toFixed(2));  }  function treePr () {  let x = -5, min = 999, argmin = 0, y;  while (x <= 5) {  y = result(x);  console.log("x = " + x.toFixed(1) + " y = " + y.toFixed(2));  x += 0.3;  if (y <= min){  min = y;  argmin = x;  }  }  console.log("Мининмум в точке x: " + argmin.toFixed(2));  }  function init () {  twoPr();  treePr();  }  init(); |

Листинг 1. Исходный код файла pr1