Теоретическая часть

Типы ошибок:

1. Синтаксическая ошибка – ошибка, нарушающая правила языка. Пример: if n == (неполное условие).

2. Логическая ошибка – код выполняется без ошибок, но результат неверный. Пример: return n + 1 вместо return n — 1.

3. Ошибка в условии – неверное условие в логическом операторе. Пример: if n < 0: return True вместо return False.

4. Ошибка обработки граничных случаев – некорректная работа на крайних значениях. Пример: неправильно обрабатывается n = 0.

5. Отклонение от алгоритма – код работает корректно, но не соответствует поставленной задаче. Пример: используется линейный поиск вместо бинарного. Методы поиска ошибок:

Визуальный анализ кода

Сравнение с описанием алгоритма

Ручное тестирование на примерах

Практическая часть

1. Найденные ошибки и отклонения:

Строка 1:

Тип: Ошибка в условии

Описание: if n <= 0: return False

Почему: По алгоритму n == 0 должно возвращать True, но тут возвращает False.

Строка 6:

Тип: Отклонение от алгоритма

Описание: high = n

Почему: По заданию верхняя граница бинарного поиска должна быть n // 2 для n > 1.

2. Подозрительные, но корректные участки:

Строки 7–15: Бинарный поиск реализован правильно, сравнение квадратов корректно.

3. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Вход | Ожидаемый результат | Фактический результат | Совпадает |
| -1 | False | False | Да |
| 0 | True | False | Нет |
| 1 | True | True | Да |
| 2 | False | False | Да |
| 3 | False | False | Да |
| 4 | True | True | Да |
| 5 | False | False | Да |
| 7 | False | False | Да |
| 9 | True | True | Да |
| 15 | False | False | Да |
| 16 | True | True | Да |
| 17 | False | False | Да |
| 1000000 | True | False | Нет |

4. Исправленная версия функции

def is\_perfect\_square(n):

if n < 0:

return False

if n == 0 or n == 1:

return True

low = 2

high = n // 2

while low <= high:

mid = (low + high) // 2

mid\_sq = mid \* mid

if mid\_sq == n:

return True

elif mid\_sq < n:

low = mid + 1

else:

high = mid — 1

return False

Исправлено условие n == 0 (должно возвращать True).

Установлен правильный диапазон high = n // 2 для бинарного поиска.

5. Анализ и выводы

Наиболее серьезная ошибка: if n <= 0 — ломает корректную работу на n = 0.

Чаще всего встречающийся тип ошибки: Отклонение от алгоритма.

Самый эффективный тест: n = 0 и n = 1000000 — выявляют основные ошибки.