

Как решать алгоритмические задачи

1. Прочитайте условие и обратите внимание на:

- то, что нужно найти;
- ограничения на входные данные;
- что дано на вход и в каком формате;
- лимиты по скорости и памяти.

2. Теперь придумайте пример:

- возьмите пример из условия задачи;
- придумайте свой пример;
- если возможно, визуализируйте пример.

3. Напишите наивный алгоритм так, чтобы он был понятен школьнику. Если это не получилось:

- расскажите, как вы находите ответ для данных из примера;
- введите дополнительные ограничения на входные данные;
- уменьшите объём входных данных;
- попробуйте поочерёдно применить уже изученные алгоритмы и структуры данных.

4. Оцените сложность алгоритма: укладывается ли ваш алгоритм в лимиты?

- укладывается по времени;
- укладывается по памяти.

5. Подумайте над более эффективной реализацией:

- проверьте код на наличие bottlenecks;
- удалите ненужные действия;
- проверьте структуры данных;
- проверьте отсутствие повторяющейся работы;
- найдите компромисс между временем и памятью, учитывая заданные ограничения на входные данные и лимиты.

6. Напишите аккуратный код. Обратите внимание на:

- имена переменных;
- имена функций;
- codestyle вашего языка.

7. Перечитайте код и проверьте:

- условные выражения;
- условия выхода из циклов и функций;
- константы.

8. Протестируйте свой код как хакер. Используйте:

- минимальный пример;
- максимальный по условию задачи пример;
- пример, содержащий особые случаи для конкретного типа данных.