

# Workshop

Урок 1



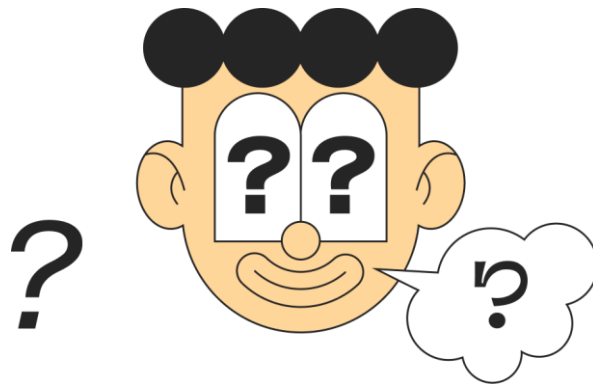


## Регламент

- ✂ Получаем задание
- ✂ Ученики разбиваются на группы в сессионные комнаты зум
- ✂ Один из учеников в группе расшаривает экран
- ✂ Выполняем задание на определенное время
- ✂ Проверяем правильность выполнения
- ✂ Переходим к новому заданию



Все ли было понятно в лекции?





# Задания





## Задание 1 (тайминг 5 минут)

Необходимо написать алгоритм, считающий сумму всех чисел от 1 до  $N$ . Согласно свойствам линейной сложности, количество итераций цикла будет линейно изменяться относительно изменения размера  $N$ .





## Задание 2 (тайминг 10 минут)

Написать алгоритм поиска простых чисел (делятся только на себя и на 1) в диапазоне от 1 до  $N$ . В алгоритме будет использоваться вложенный `for`, что явно говорит о квадратичной сложности, на этом стоит акцентировать внимание





## Задание 3 (тайминг 15 минут)

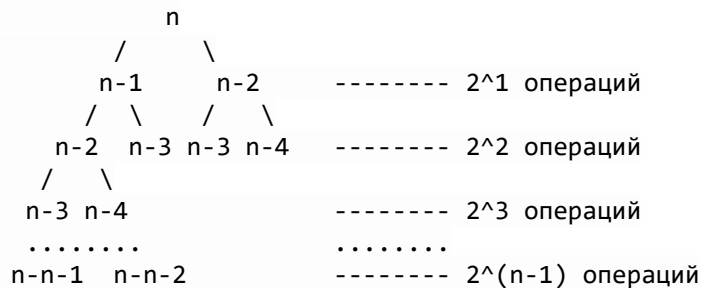
1. Необходимо написать алгоритм поиска всех доступных комбинаций (посчитать количество) для количества кубиков  $K$  с количеством граней  $N$ .
2. У вас есть 2 варианта на выбор – количество кубиков может быть строго ограничено (4 кубика, например), либо их количество будет динамическим. Выбор за вами.
3. Если вы реализуете простой вариант, обращает внимание, что данное решение имеет сложность  $O(n^4)$ , но если количество кубиков сделать переменной, то она трансформируется в  $O(n^k)$ , что будет представлять собой экспоненциальную сложность. Для второго решения очевидно, что его сложность  $O(n^k)$  с самого начала.





## Задание 4 (тайминг 10 минут)

1. Пишем алгоритм поиска нужного числа последовательности Фибоначчи.
2. Считаем, что 1 и 2 значения последовательности равны 1.
3. Искать будем по формуле  $O_n = O_{n-1} + O_{n-2}$  что предполагает использовать рекурсивного алгоритма.







## Задание 5 (тайминг 10 минут)

1. Пишем алгоритм поиска нужного числа последовательности Фибоначчи, но в этот раз откажемся от рекурсии и воспользуемся обычным алгоритмом, что даст нам линейную сложность, т.к. вычисление каждого из чисел последовательности будет происходить ровно 1 раз.

1. Вариантов решения может быть несколько, но нужно предложить студентам считать последовательность с первого числа и далее до тех пор, пока не доберемся до нужного номера. При этом значение предыдущих элементов нужно сохранять, чтобы не приходилось вычислять заново, как это происходило при рекурсивном методе.





## Задание 6 (тайминг 10 минут)

Необходимо сравнить скорость работы 2 алгоритмов вычисления чисел Фибоначчи и определить, какой из них работает быстрее. Так различия начнутся уже с 40 члена последовательности.





Итоги






Что было  
сложного на  
семинаре?





Напишите 3 вещи в  
комментариях, которым  
вы научились сегодня.



**Спасибо**   
**за внимание**

