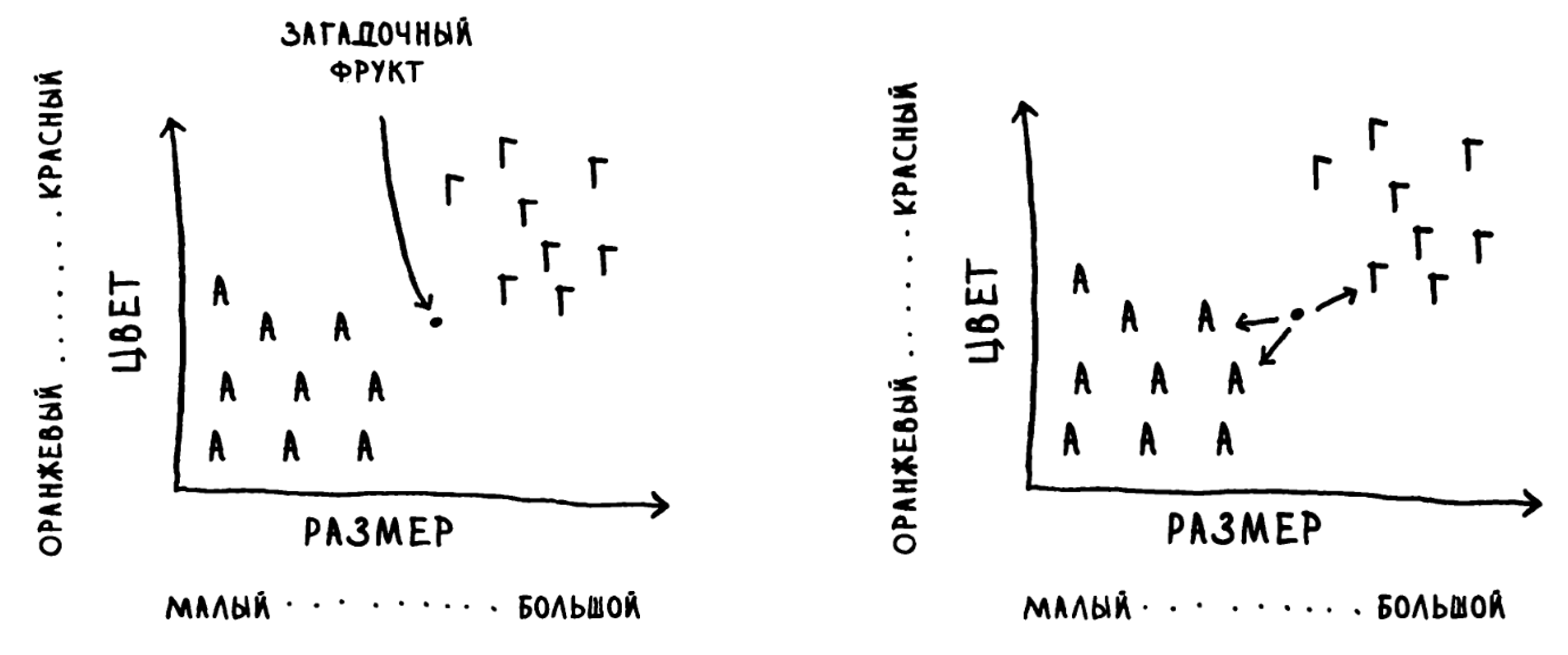
**Задание 1.** Вы создали новый алгоритм майнинга криптовалюты 😊. Для реализации этого алгоритма вам нужна функция Factorization(). Она получает в качестве аргумента число N типа BigInteger и возвращает набор чисел BigInteger – это все простые множители входного числа N (то есть это простые числа, и, если их все перемножить – получится число N). Если число N меньше двух – генерируется исключительная ситуация.

1. Написать и протестировать функцию Factorization(). Многопоточность не использовать.
2. Превратить функцию Factorization() в асинхронную функцию FactorizationAsync() при помощи TaskCompletionSource. Пусть при этом асинхронная работа реализуется при помощи потока Thread.
3. Если у нас есть два набора простых множителей двух чисел, мы можем найти наибольший общий делитель (НОД) этих двух чисел. Вспоминаем школу (<http://spacemath.xyz/nod_i_nok/>) или ищем алгоритм в интернете. Напишите асинхронную функцию (с модификатором async) для нахождения НОД двух чисел типа BigInteger, которая будет использовать в своей работе функцию FactorizationAsync().

**Задание 2.** *Метод k-ближайших соседей* – один из простейших алгоритмов машинного обучения (<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>).

Суть алгоритма: имеется набор объектов с известными **названиями** и числовыми **характеристиками**. Эти характеристики дают возможность описать объект как точку в -мерном пространстве. Пусть есть объект, у которого заданы характеристики, но нет названия. Чтобы найти название этого объекта, найдём ближайших к нему известных объектов. То название, которое будет у ближайших соседей повторяться чаще, сделаем названием нашего неизвестного объекта (алгоритм классификации).

Пример: апельсины и грейпфруты, . Неизвестный объект оказался апельсином.



1. Необходимо реализовать алгоритм метода k-ближайших соседей в двумерном пространстве характеристик. Вход алгоритма: набор известных объектов, две характеристики неизвестного объекта и значение . Каждый объект состоит из строкового названия и двух (для простоты) числовых характеристик (можно использовать кортежи). Выход алгоритма: строка с названием неизвестного объекта. Постарайтесь при реализации алгоритма использовать возможности класса Parallel или технологии PLINQ.
2. Создайте консольное приложение для тестирования алгоритма. Консольное приложение читает входные данные из текстового файла. Пример файла приведён ниже. Полный файл – в приложении к заданию.

**Пример файла с данными**

orange,1.0,1.1

orange,1.0,2.0

orange,4.0,2.0

orange,4.0,3.2

orange,2.0,4.3

orange,3.0,4.1

orange,3.0,3.1

grapefruit,6.2,8.5

grapefruit,7.0,7.0

grapefruit,7.2,8.0

grapefruit,8.0,7.0

grapefruit,8.2,8.0

grapefruit,8.0,9.0

grapefruit,8.5,8.6

**Задание 3.** CSV-файл data.csv содержит информацию об отпусках сотрудников компании. Каждая строка – это имя сотрудника, дата начала отпуска, дата окончания отпуска. Фрагмент файла:

Dlsw Fmbnaswsg,1/25/2016,2/3/2016

Iqbidtaw Ndxqjdpsg,6/8/2015,6/19/2015

Lrnywj Fdgyzw,6/8/2015,6/16/2015

Имена сотрудников зашифрованы. Полный файл – в приложении к заданию.

1. Прочитайте содержимое файла в коллекцию объектов в памяти. Если какая-то строка файла содержит ошибку, строка игнорируется, информация об ошибке и номере строки выводится на консоль, но чтение файла не прекращается.
2. На основании данных об отпусках сформируйте набор объектов, где каждый объект хранит имя сотрудника и список его отпусков (отпуск – эта пара дат start и end). Нужно сохранить только данные об отпусках, которые полностью прошли во второй половине 2016 года. Если у сотрудника таких отпусков не было – сотрудник игнорируется.
3. Запишите сформированный набор объектов в JSON-файл persons.json. Желаемый вид итогового JSON-файла поясняет фрагмент ниже.

[

{

"name": "Pdgbq Rsqpdfmbg",

"vacations": [

{

"start": "2016-07-27T00:00:00",

"end": "2016-08-07T00:00:00"

},

{

"start": "2016-09-22T00:00:00",

"end": "2016-09-22T00:00:00"

}

]

},

{

"name": "Hdxyjd Zbtfmnp",

"vacations": [

{

"start": "2016-07-17T00:00:00",

"end": "2016-07-29T00:00:00"

},

]

},