

Лабораторная 11

Сдать до 23.11

Задание 1 [Обязательное задание] Реализуйте кэш для хранения объектов одного типа (10 баллов)

В кэше могут храниться объекты одного типа, реализующего интерфейс `IDisposable`.

Каждый объект имеет временную метку, когда к нему было последнее обращение из кэша. Также объекты в кэше (теоретически) могут обращаться к неуправляемым ресурсам, поэтому нужно гарантировать их освобождение.

У кэша можно задавать максимальный размер. Когда поступает запрос на добавление нового элемента в кэш, но места больше нет (достигнут максимальный размер кэша – т.е. уже содержится максимальное количество элементов), то кэш принудительно запускает очистку – уничтожаются все «старые» объекты.

«Старые» объекты определяются так: если к объекту не было обращений дольше, чем задано (интервал времени можно задавать параметром при создании кэша), то объект удаляется из кэша и должен быть уничтожен.

Также кэш должен регистрироваться на получение уведомлений о приближении полной сборки мусора при помощи метода **`RegisterForFullGCNotification`**. И, в случае получения уведомления, должен запустить очистку кэша.

Предложите свою реализацию кэша и класса объектов, которые в нем хранятся. Объекты в кэше должны корректно уничтожать занимаемые ресурсы.

Реализуйте пример (или серию примеров), в которых происходит запуск очистки кэша в двух сценариях:

1. после добавления элемента (5 баллов)
2. при приближении полной сборки мусора (5 баллов)

Задание 2. Генерация перестановок (4 балла)

Создайте функцию для генерации всех перестановок строковых символов.

Примеры

`Permutations("AB") → "AB BA"`

`Permutations("CD") → "CD DC"`

`Permutations("EF") → "EF FE"`

`Permutations("NOT") → "NOT NTO ONT OTN TNO TON"`

`Permutations("RAM") → "AMR ARM MAR MRA RAM RMA"`

`Permutations("YAW") → "AWY AYW WAY WYA YAW YWA"`

Отсортируйте результаты в алфавитном порядке, прежде чем возвращать их.

Задание 3. Автобусные маршруты (4 балла)

Вам предоставляется массив маршрутов автобусов, где `routes[i]` — это автобусный маршрут, который *i*-й автобус повторяет все время в цикле.

Например, если `routes[0] = [1, 5, 7]`, это означает, что 0-й автобус едет в последовательности 1 -> 5 -> 7 -> 1 -> 5 -> 7 -> 1 ->... все время.

Вы начинаете с одной автобусной остановки. Вам нужно добраться до заданной другой остановки, которая может принадлежать маршруту другого автобуса. Перемещаться между автобусными остановками возможно только на автобусах.

Определите наименьшее количество автобусов, на которые вам необходимо будет сесть, чтобы добраться от исходной остановки до целевой.

Пример 1:

Вход: маршруты = `[[1,2,7],[3,6,7]]`, старт = 1, цель = 6

Выход: 2

Пояснение: Лучшая стратегия - сесть на первый автобус до автобусной остановки 7, затем сесть на второй автобус до автобусной остановки 6.

Пример 2:

Вход: маршруты = `[[7,12],[4,5,15],[6],[15,19],[9,12,13]]`, старт = 15, цель = 12

Выход: -1

Вы не сможете добраться из 15 в 12, используя указанные маршруты