Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №1**по курсу «Алгоритмы и структуры данных»

Выполнил:

Студент группы P3212  
Анищенко Анатолий Алексеевич

Преподаватель:

Косяков М.С.

г. Санкт-Петербург  
2020 г.

**Задача №2025 «Стенка на стенку»**

**Пояснение к примененному алгоритму:**

Каждый боец за время потехи участвует по разу в схватке с каждым, кто не состоит в его команде. Чтобы в итоге было как можно больше схваток, каждый боец должен биться с как можно большим количеством участников. Иными словами, должно быть как можно меньше бойцов, с которыми он не бьётся, то есть, в каждой команде должно быть как можно меньше бойцов. Исходя из этого, равномерно «размазываем» участников по командам, а если N (кол-во бойцов) не кратно K (кол-ву команд), то оставшихся также распределяем поровну. Затем просто формулами считаем количество схваток.

Предположим, разбили команды таким образом, что задействовано не максимальное количество команд. Тогда мы можем разбить какую-то команду (в которой больше одного человека) на 2 команды без изменения количество человек. Количество боев увеличилось, так как количество боёв между участниками этих команд и остальных не менялось (оно зависит только от количество игроков), а количество игры между получившимися двумя командами из 0 превратилось в какое-то число, то есть увеличилось, ч т. д.

Предположим, игроки разбиты не поровну, тогда существуют команды, в которых количество участников отличается минимум на 2. Положим в первой команде n человек, тогда во второй n+x (x >= 2) человек. Докажем, что перенос человек из большей команды в меньшую увеличит количество боев (сравнив бои между командами, количество боёв между этими и другими командами очевидно не поменяются):

n(n + x) vs (n + x / 2)(n + x / 2)

n^2 + xn vs n^2 + xn + x^2 / 4

0 vs x^2 / 4

0 < x^2 / 4 (т.к. x >= 2)

Верно, значит количество боев увеличилось, при этом количество команд и игроков осталось неизменным.

**Задача №1005 «Куча камней»**

**Пояснение к примененному алгоритму:**

Так как количество камней ограничено всего 20 камнями, то можно перебрать все варианты. Используем битовую маску (0 – первая кучка, 1 – вторая кучка). После набора кучек проверим на минимальность разность.

**Задача №1155 «Дуоны»**

**Пояснение к примененному алгоритму:**

Заметим, с помощью не сложных математических выкладок можно доказать, что задача имеет решение только в случае, если сумма дуонов на несмежных вершинах совпадает, т. е. A+C+F+H = B+D+E+G.

Обнуляем сначала вершину А, аннигилируя все дуоны на смежных с ней вершинах. Если дуоны на вершине А не обнулились, а дуоны на смежных вершинах закончились, то добавляем пары дуонов на ребро таким образом, чтобы один дуон был смежный с вершиной А. После аннигилируем эти пары дуонов.

Дальше тоже самое делаем с вершинами смежными с A (B, D, E). После чего у нас остаётся всего 4 вершины: C, F, H из первой группы и G из второй группы. Так как решение возможно (мы проверили это в самом начале) и мы аннигилировали вершины одинаково и первой и второй группе одинаково, следовательно сейчас вершина G равна сумме вершин C, F, H. Просто проаннигилируем вершину G и решим задачу)

**Задача №1296 «Гиперпереход»**

**Пояснение к примененному алгоритму:**

Начиная с первого элемента прибавляем его к текущей сумме. Если результат больше 0, то в дальнейшем он может нам только увеличить результат. Если текущая сумма меньше 0, то пользы она нам не принесет и начинаем с этого момента новый прыжок (т. е. обнуляем текущую сумму). На каждой итерации проверяем, максимален ли результат, если да, то сохраняем. Таким образом всегда есть актуальный максимальный результат.

**Задача №1401 «Игроки»**

**Пояснение к примененному алгоритму:**

Заметим, что для квадрата 2 х 2 задача решена. Докажем, что мы можем решить данную задачу для любых квадратов размера 2^n x 2^n. Допустим мы умеем решать задачу для 2^n x 2^n, решим для 2^(n + 1) x 2^(n + 1).

Во-первых, разобьём наш квадрат на 4 квадрата размера 2^n x 2^n. Один из получившихся квадратов содержит в себе дырку и его мы можем разрезать согласно предположению. Докажем, что оставшиеся квадраты мы также можем разрезать, возьмём оставшийся треуголничик из центра большого квадрата (2^(n + 1) x 2^(n + 1)) отдельно. Теперь в оставшихся 3ёх маленьких квадратах, у нас ровно по одной вырезанной точки, а такую задачу мы решать умеем.

Основной принцип, используемый при решении задачи – рекурсия.

**Исходный код:**

<https://github.com/AnatoliiAnishchenko/Algo_and_DS/tree/master/Part1>