**Вологда, Импульс, смена по информатике, декабрь 2020**

**Олимпиада закрытия**

**Задача A – Гусеница**

 Несложная задача на вывод формулы.

**Задача B – Гири**

 Определим, когда решения нет. Пусть *x* – вес первой пары. Всего n пар, вес каждой следующей на единицу больше предыдущей: *x*, *x*+1, ..., *x*+*n*-1. Тогда по формуле арифметической прогрессии сумма всех пар равна (2*x*+*n*-1)\**n*/2. Она также равна сумме всех гирь: (1+2\**n*)\*n, то есть (2*x*+*n*-1)\**n*/2 = (1+2\**n*)\**n*. Выражаем *x*: *x* = (3+3*n*)/2

 Вывод: чтобы x было целым, нужно, чтобы *n* было нечётным. При чётном *n* решения нет.

 Алгоритм построения проиллюстрируем на примере для n=5. Вначале возьмём пары <1, 8>, <2, 9>, <3, 10>. А затем в промежутки между ними вставим пары <4, 6>, <5, 7>. Получился ответ: <1, 8>, <4, 6>, <2, 9>, <5, 7>, <3, 10>.

**Задача C – Максимумы в квадратах**

 Задача на динамическое программирование. Пусть f(i,j) – ответ для элемента с индексами i, j (где i < j) – то есть, максимум в таком квадрате:

 **i** **j**

**i** .....

 .....

**j** .....

 Заметим, что f(i, j-1) – это максимум в этом квадрате без последней строки и столбца, а f(i+1, j) – максимум в этом квадрате без первой строки и столбца. Объединение же их почти полностью покрывает наш квадрат – неохваченными остаются лишь два элемента в левом нижнем и правом верхнем углу. Поэтому:

f(i, j) = max(f(i, j-1), f(i+1,j), a[j][i], a[i][j])

 Для случая i > j можно просто поменять в формуле индексы местами.

**Задача D – Цивилизация**

 Задачу можно решить с помощью алгоритма Дейкстры (вариант с очередью с приоритетами или множеством для быстрого поиска вершины с наименьшей оценкой).

 Также можно свести задачу к поиску в ширину – для этого достаточно каждую клетку, в которой растёт лес, превратить в две вершины графа – входную и выходную, а между ними провести дополнительное ребро.

**Задача E – Мисс Крамплботтом – гондольер**

С помощью алгоритм Краскала строим максимальное остовное дерево (вместо минимального) и смотрим, сколько ребер в полученном дереве нуждаются в расширении.

**Задача F – Супер простые числа**

Можно заметить, что:

1) Все такие числа состоят из цифр 2, 3, 5, 7 (так как подстроки длины 1 должны быть простыми).

2) В таких числах не может быть 2 и 5 не на первом месте

3) И не может двух подряд идущих одинаковых

4). Числа, начинающиеся с двойки – это 2 и 23. Других таких чисел нет, так как 237 и 273 - не простые.

5). Числа, начинающиеся с пятерки – это 5 и 53. Других таких чисел нет, так как 57 и 537 - не простые.

6). Числа, начинающиеся с тройки – это 3, 37, 373. Других таких чисел нет, так как 3737 – не простое.

7). Числа, начинающиеся с семерки – это 7, 73. Других таких чисел нет, так как 737 – не простое.

Итого, суперпростыми числами являются только следующие:

2 3 5 7 23 37 53 73 373.

В решении нужно лишь сосчитать, сколько из них не превосходят n.

**Задача G – Максимальный XOR**

 Заносим числа в бор. При поиске наилучшего числа делаем следующее. Если у нас текущая цифра – ноль, то в боре пытаемся идти в единицу (если есть куда), а если единица – то идти в ноль. Если такого ребра нет, то идём по единственному ребру.