

Введение

В данном отчёте представлено подробное описание решение Задачи №2.

Схема решения, описание алгоритма

В задаче сказано, что у студентов есть 2 дня до праздников, соответственно у них есть 2 попытки на то, чтобы определить где находится «Золотая хвостовка». Так как количество испытуемых должно быть максимальным, то их надо переиспользовать. Введём параметры, которые мы будем искать

x_1 — количество человек, необходимых для первого дня

x_2 — количество человек, необходимых для второго дня

Если представить, что x_1 человек участвовали в первом дне, то можно сократить количество рассматриваем судных с 80 до $80/x_1$. Например, если $x_1 = 4$, то количество рассматриваемых сундуков через день будет всего 20 (один из студентов точно получит «Золотую хвостовку»). Тогда свяжем с тем, что имеем x_2

$$\begin{cases} x_2 = \frac{80}{x_1} \\ \max(x_2, x_1) - \min \text{ среди всех } \max(x_2, x_1) \end{cases}$$

Мы должны найти такие параметры, что максимальное из них (требуемое кол-во студентов в первый и второй день) будет минимальным среди всех подходящих.

Доказательство оптимальности полного решения

Увидим, что до некоторого x_1 и x_2 , то $\max(x_2, x_1)$ убывает, а после возрастает. Сами значения максимальные значения «отзеркалены», потому что x_1 и x_2 можно переставить местами и решение не поменяется. По смыслу получается, что нам без разницы x человек будут в первый день или во второй, потому что нас интересует максимум за два дня.

x1	x2	max
2	40	40
4	20	20
5	16	16
8	10	10
10	8	10
16	5	16
20	4	20
40	2	40

Программный код

Github с решением (файл task2.py)

<https://github.com/andreyvydra/olimp>

Программный код представлен на ЯП Python

```
min = 100
x, y = 0, 0
for x1 in range(1, 80):
    for x2 in range(1, 80):
        if x2 == 80 / x1:
            if max(x1, x2) < min:
                min = max(x1, x2)
                x = x1
                y = x2
print("Необходимое кол-во человек:", min)
print(f"{x} в первый день, {y} во второй день")
```

Ответ

Минимальное кол-во испытателей – 10.

Вывод

В данной задаче я нашёл алгоритм поиска оптимального количества испытуемых. Реализовал данный алгоритм с помощью ЯП Python.