

Лабораторная работа 3

Вариант 29

1. Формулировка проблемы

Задача 29. Расселение

Руководство фирмы ООО «Вектор», состоящей из двух отделов, решило на празднование годовщины основания организовать поездку всех сотрудников в санаторий. Санаторий располагает одноместными и двухместными номерами. Для укрепления корпоративного духа было принято решение селить в двухместные номера сотрудников из разных отделов. Если в одном из отделов больше людей, чем в другом, то тех, кому не хватило пары, расселяют в одноместные номера. В силу разного возраста сотрудников вводится показатель недовольства, равный разности в возрасте заселяемых в один номер (вычисляется для каждого номера, а не сотрудника). Для тех, кто попадает в одноместный номер, он равен половине возраста.

Формат входных данных

В первой строке задаются возрасты сотрудников первого отдела, разделённые пробелами. Во второй строке — возрасты сотрудников второго отдела. Возраст сотрудника должен лежать в интервале от 18 до 60.

Формат выходных данных

Единственная строка должна содержать наименьший суммарный показатель недовольства (вычисляется как сумма показателей всех номеров). Число должно иметь одну цифру после точки.

Пример

| входной файл | выходной файл |
|----------------|---------------|
| 20 24 18 42 | 20.0 |

2. Формула для расчёта недовольства

Недовольство для двухместных номеров:

- Если сотрудники размещены в паре, недовольство рассчитывается как разность возрастов:

$$D_{pair} = |A_i - B_i|$$

где A_i и B_i — возраста сотрудников из разных отделов, размещённых в одной паре.

Недовольство для одноместных номеров:

- Если сотрудник размещен в одноместном номере, недовольство рассчитывается как половина возраста:

$$D_{single} = \frac{A_k}{2}$$

где A_k — возраст сотрудника, размещенного в одноместном номере.

Общее недовольство определяется как сумма недовольств для всех номеров:

$$D_{total} = \sum D_{pair} + \sum D_{single}$$

3. Реализация на языке Python

```
def calculate_dissatisfaction(department1, department2):
    # Сортируем возраста сотрудников
    department1.sort()
    department2.sort()

    # Определяем количество сотрудников в каждом отделе
    len1 = len(department1)
    len2 = len(department2)

    # Определяем количество пар и одноместных номеров
    pairs = min(len1, len2)
    single_rooms1 = len1 - pairs
    single_rooms2 = len2 - pairs

    # Расчет недовольства для двухместных номеров
    dissatisfaction = 0.0
    for i in range(pairs):
        dissatisfaction += abs(department1[i] - department2[i])

    # Расчет недовольства для одноместных номеров
    for i in range(single_rooms1):
        dissatisfaction += department1[pairs + i] / 2.0
    for i in range(single_rooms2):
        dissatisfaction += department2[pairs + i] / 2.0

    return round(dissatisfaction, 1)

# Ввод данных
department1 = list(map(int, input().strip().split()))
department2 = list(map(int, input().strip().split()))
```

```
# Вычисление и вывод результата
result = calculate_dissatisfaction(department1, department2)
print(result)
```

4. Объяснение кода

Определение функции

```
def calculate_dissatisfaction(department1, department2):
```

- Входные параметры:
 - department1: Список возрастов сотрудников первого отдела.
 - department2: Список возрастов сотрудников второго отдела.

Сортировка списков

```
department1.sort()
department2.sort()
```

- Сортируем оба списка возрастов по возрастанию. Это упрощает процесс формирования пар, так как мы можем сопоставлять сотрудников с близкими по возрасту.

Определение размеров списков

```
len1 = len(department1)
len2 = len(department2)
```

- Определяем количество сотрудников в каждом отделе. Это необходимо для вычисления количества пар и одноместных номеров.

Вычисление количества пар и одноместных номеров

```
pairs = min(len1, len2)
single_rooms1 = len1 - pairs
single_rooms2 = len2 - pairs
```

- Логика:
 - pairs: Количество пар, которые можно сформировать, равно минимальному значению между количеством сотрудников в двух отделах. Это гарантирует, что мы не создадим больше пар, чем сотрудников в одном из отделов.
 - single_rooms1: Количество сотрудников в первом отделе, которые не смогли образовать пару и будут размещены в одноместных номерах.
 - single_rooms2: Количество сотрудников во втором отделе, которые не смогли образовать пару и будут размещены в одноместных номерах.

Расчет недовольства для двухместных номеров

```
dissatisfaction = 0.0
for i in range(pairs):
    dissatisfaction += abs(department1[i] - department2[i])
```

- Логика:
 - Инициализируем переменную dissatisfaction для хранения общего показателя недовольства.
 - В цикле перебираем пары сотрудников из обоих отделов. Для каждой пары вычисляем разность их возрастов с помощью функции abs(), которая возвращает абсолютное значение разности.
 - Суммируем все недовольства для двухместных номеров.

Расчет недовольства для одноместных номеров

```
for i in range(single_rooms1):
    dissatisfaction += department1[pairs + i] / 2.0
for i in range(single_rooms2):
    dissatisfaction += department2[pairs + i] / 2.0
```

- Логика:
 - Для сотрудников в одноместных номерах (которые не смогли образовать пару) мы добавляем к общему недовольству половину их возраста.
 - Первый цикл обрабатывает сотрудников из первого отдела, которые размещены в одноместных номерах.
 - Второй цикл обрабатывает сотрудников из второго отдела.

Возврат результата

```
return round(dissatisfaction, 1)
```

- Возвращаем итоговое значение недовольства, округленное до одной цифры после запятой. Это позволяет обеспечить требуемую точность вывода.

5. Результат

```
PS C:\Users\tshyb\source\repos\4th\7sem\ISO\lab3> python -u
20 24
18 42
20.0
PS C:\Users\tshyb\source\repos\4th\7sem\ISO\lab3> python -u
20 30
25 35
10.0
PS C:\Users\tshyb\source\repos\4th\7sem\ISO\lab3> python -u
18 20 22 24 26
19 21 23 25 27
5.0
PS C:\Users\tshyb\source\repos\4th\7sem\ISO\lab3> python -u
25 30
27.5
```