**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**)

**Институт управления и цифровых технологий**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**ОТЧЕТ ПО лабораторной работЕ №7**

**«Кластеризация МЕТОДОМ K-СРЕДНИХ И МЕТОДОМ ПОСЛЕДОВАТЕЛЬНЫХ СЛИЯНИЙ»**

**ПО дисциплине** **«Нейроинформатика»**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:   
студент группы УИБ-311

Москаленко В.А.

Проверил:

Доцент Малинский С.В.

(должность, ФИО)

МОСКВА 2024

СОДЕРЖАНИЕ

[1 ЗАДАНИЕ 3](#_Toc181742767)

[2 ТЕОРЕТИЧЕСКАЯ ИНФОРМАЦИЯ О МЕТОДАХ КЛАСТЕРИЗАЦИИ 4](#_Toc181742768)

[2.1 Метод последовательных слияний 4](#_Toc181742769)

[2.2 Метод k-средних 4](#_Toc181742770)

[3 КЛАСТЕРИЗАЦИЯ МЕТОДОМ ПОСЛЕДОВАТЕЛЬНЫХ СЛИЯНИЙ И ПОСТРОЕНИЕ КРИВОЙ ТОРНДАЙКА 6](#_Toc181742771)

[4 ДЕЛЕНИЕ НА КЛАСТЕРЫ МЕТОДОМ K-СРЕДНИХ 8](#_Toc181742772)

[ЗАКЛЮЧЕНИЕ 9](#_Toc181742773)

[ПРИЛОЖЕНИЕ А 10](#_Toc181742774)

[ПРИЛОЖЕНИЕ Б 11](#_Toc181742775)

[ПРИЛОЖЕНИЕ В 12](#_Toc181742776)

**1 ЗАДАНИЕ**

Решить задачу кластерного анализа для обучающей выборки из объектов с помощью метода k-средних и метода последовательных слияний. Составить графики с результатами кластеризации обоими методами, построить кривую Торндайка.

Вариант – 3643810

Выбранный набор признаков – 3,4

**2 ТЕОРЕТИЧЕСКАЯ ИНФОРМАЦИЯ О МЕТОДАХ КЛАСТЕРИЗАЦИИ**

## **2.1 Метод последовательных слияний**

Метод последовательных слияний кластерного анализа представляет собой подход к группировке данных, где процесс начинается с того, что каждый объект рассматривается как отдельный кластер. Постепенно кластеры объединяются на основе минимального расстояния между ними, продолжаясь до тех пор, пока не будет достигнуто необходимое число кластеров.

По мере слияния в кластерах увеличивается количество объектов, а среднее расстояние между центроидами этих кластеров увеличивается. На каждом этапе фиксируется расстояние между объединяемыми кластерами, что позволяет построить график изменения расстояний — кривая Торндайка. Значительные скачки на ней указывают на слияние различающихся групп. Анализ таких скачков помогает определить оптимальное число кластеров.

## **2.2 Метод k-средних**

Метод k-средних — это один из самых распространённых алгоритмов кластеризации, разработанный ещё в середине XX века. Основной смысл метода заключается в оптимизации: сначала случайно выбираются центроиды кластеров, а затем объекты распределяются по кластерам на основе минимального расстояния до этих центроидов.

После начальной классификации для каждого кластера пересчитывается центроид. На следующем шаге точки снова распределяются по кластерам в соответствии с новыми центрами. Этот процесс повторяется до тех пор, пока изменение в позициях центроидов и составе кластеров не прекратится.

Алгоритм k-средних обычно быстро достигает стабильного состояния за сравнительно небольшое число итераций. Однако качество кластеризации может зависеть от начального выбора центроидов, что делает метод чувствительным к начальным условиям.

# **3 КЛАСТЕРИЗАЦИЯ МЕТОДОМ ПОСЛЕДОВАТЕЛЬНЫХ СЛИЯНИЙ И ПОСТРОЕНИЕ КРИВОЙ ТОРНДАЙКА**

Для вычисления расстояний между объектами использовалось следующая формула (1):

(1)

Метод был использован для каждого количества кластеров, начиная от 60 кластеров и заканчивая 2 кластерами. Была построена кривая Торндайка, она изображена на рисунке 1.

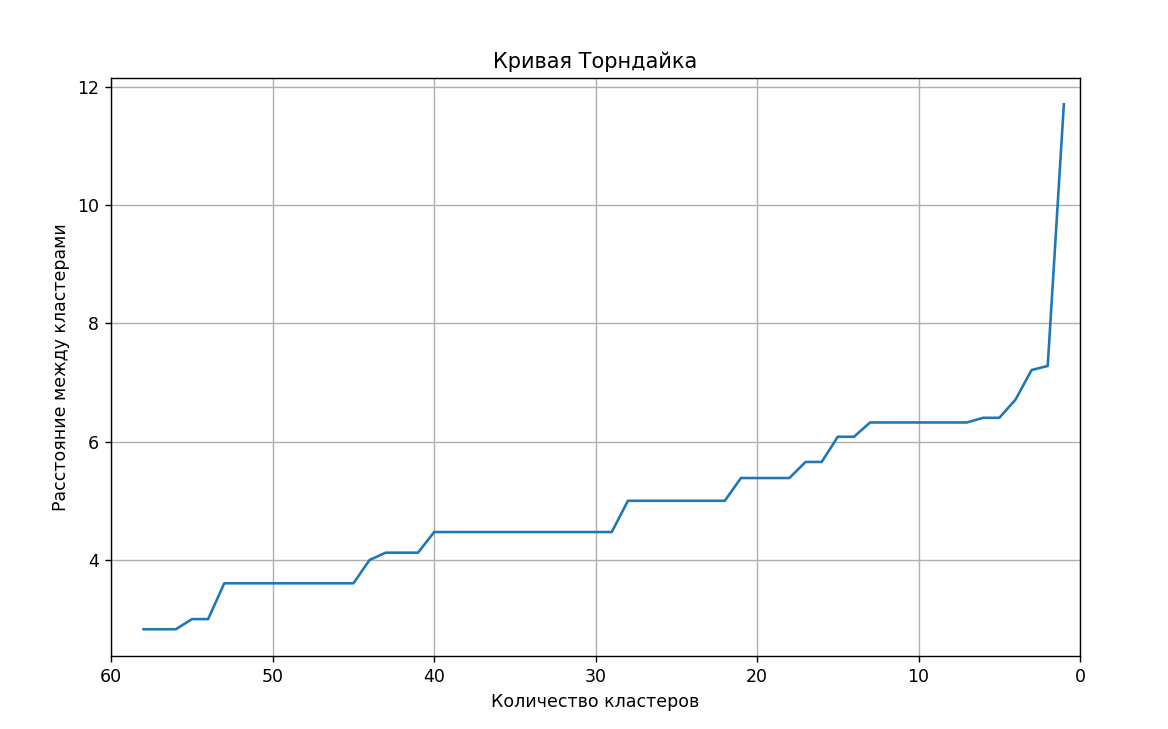


Рисунок 1 – Кривая Торндайка

Анализируя кривую, было выбрано оптимальное количество кластеров, равное 4. График разделения объектов методом последовательных слияний представлен на рисунке 2.

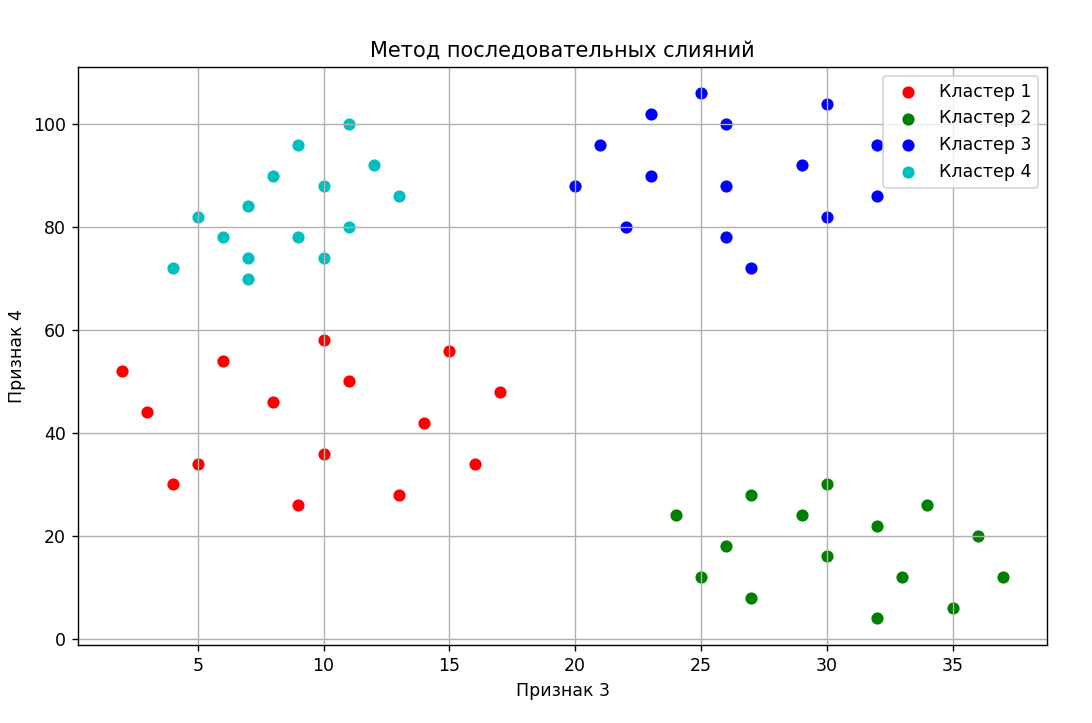


Рисунок 2 – Кластеризация методом последовательных слияний

# **4 ДЕЛЕНИЕ НА КЛАСТЕРЫ МЕТОДОМ K-СРЕДНИХ**

По кривой Торндайка было выбрано оптимально количество кластеров для разделения – 4 кластера. График разделения объектов k-средних представлен на рисунке 3.

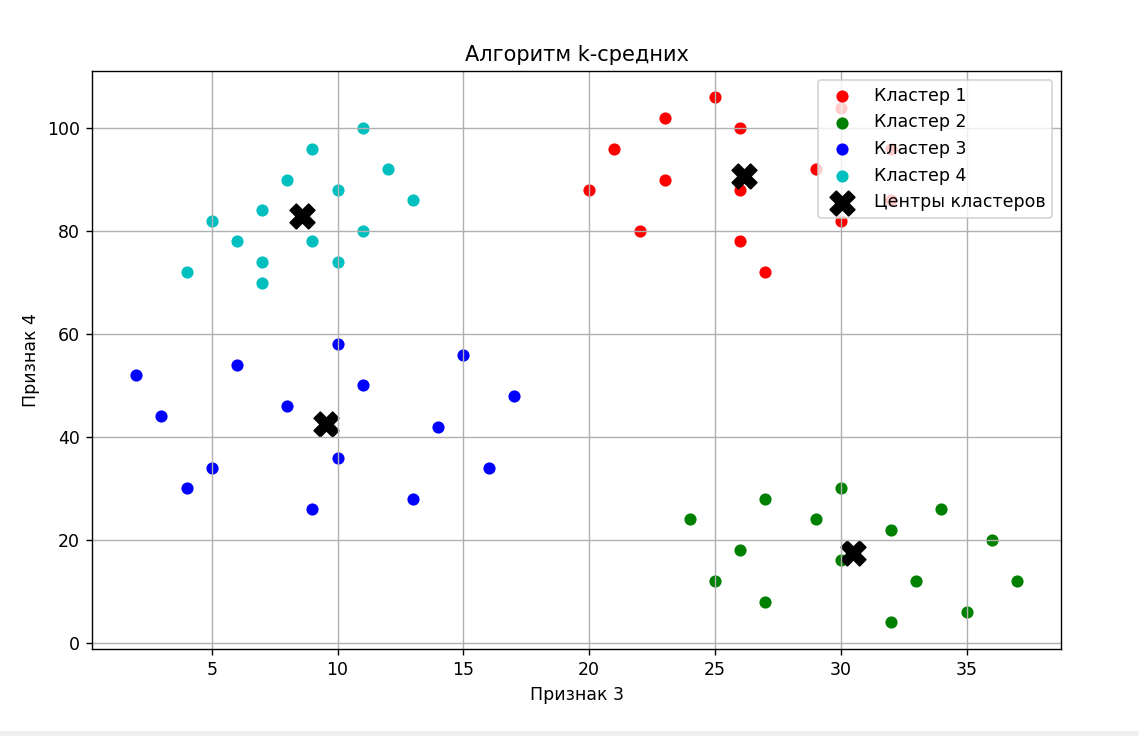


Рисунок 3 – Кластеризация методом k-средних

# **ЗАКЛЮЧЕНИЕ**

В результаты проделанной работы задача кластерного анализа была решена методом последовательных слияний и методом k-средних. Анализ результатов показал идентичную кластеризацию обоими методами. Была построена кривая Торндайка, с помощью которой было определено оптимальное количество кластеров – 4. Результаты показали правильность выбора этого количества. Задача кластерного анализа для разбиения объектов была решена корректно.

# **ПРИЛОЖЕНИЕ А**

import numpy as np

import matplotlib.pyplot as plt

# Данные

dannie = """

2 52

3 44

4 30

5 34

6 54

8 46

9 26

10 36

10 58

11 50

13 28

14 42

15 56

16 34

17 48

24 24

25 12

26 18

27 8

27 28

29 24

30 16

30 30

32 4

32 22

33 12

34 26

35 6

36 20

37 12

20 88

21 96

22 80

23 90

23 102

25 106

26 78

26 88

26 100

27 72

29 92

30 82

30 104

32 86

32 96

7 70

4 72

7 74

10 74

6 78

9 78

11 80

5 82

7 84

13 86

10 88

8 90

12 92

9 96

11 100

"""

matrica=np.array([list(map(int, line.split())) for line in dannie.strip().split("\n")])

def PoiskDistantMatrica(matrica):

n=len(matrica)

distantMatrica=np.zeros((n, n))

for i in range(n):

for j in range(n):

if i!=j:

distantMatrica[i][j]=np.sqrt(np.sum((matrica[i]-matrica[j])\*\*2))

else:

distantMatrica[i][j]=np.inf

return distantMatrica

def PoiskMinDistant(distantMatrica):

minDist=np.inf

minIndixes=(-1, -1)

n=distantMatrica.shape[0]

for i in range(n):

for j in range(i+1, n):

if distantMatrica[i][j]<minDist:

minDist=distantMatrica[i][j]

minIndixes=(i, j)

return minIndixes, minDist

def ObnovMatrica(distantMatrica, cluster1, cluster2):

n=distantMatrica.shape[0]

for i in range(n):

if i!=cluster1 and i!=cluster2:

distantMatrica[cluster1][i]=distantMatrica[i][cluster1]=min(distantMatrica[cluster1][i], distantMatrica[cluster2][i])

distantMatrica=np.delete(distantMatrica, cluster2, axis=0)

distantMatrica=np.delete(distantMatrica, cluster2, axis=1)

return distantMatrica

def MetodSliyaniy(matrica, KolvoClusters):

Сlusters=[[i] for i in range(len(matrica))]

distantMatrica=PoiskDistantMatrica(matrica)

Distances=[]

Distances.append(np.max(distantMatrica))

while len(Сlusters)>KolvoClusters:

(cluster1, cluster2), Mindist=PoiskMinDistant(distantMatrica)

Distances.append(Mindist)

Сlusters[cluster1].extend(Сlusters[cluster2])

del Сlusters[cluster2]

distantMatrica=ObnovMatrica(distantMatrica, cluster1, cluster2)

return Distances, Сlusters

VseDistant, FinalClasters = MetodSliyaniy(matrica, KolvoClusters=2)

VseDistant4Clasters, Clasters4 = MetodSliyaniy(matrica, KolvoClusters=4)

plt.figure(figsize=(10, 6))

plt.plot(range(1, len(VseDistant)+1), VseDistant, linestyle='-')

plt.xlabel('Количество кластеров')

plt.ylabel('Расстояние между кластерами')

plt.title('Кривая Торндайка')

OrigrTiks = plt.xticks()[0]

NewTiks = OrigrTiks[::-1]

plt.xticks(OrigrTiks, NewTiks.astype(int))

plt.xlim(0, 60)

plt.grid(True)

plt.show()

plt.figure(figsize=(10, 6))

colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']

for cluster\_idx, cluster in enumerate(Clasters4):

cluster\_points = matrica[cluster]

plt.scatter(cluster\_points[:, 0], cluster\_points[:, 1], color=colors[cluster\_idx % len(colors)], label=f'Кластер {cluster\_idx + 1}')

plt.xlabel('Признак 3')

plt.ylabel('Признак 4')

plt.title('Метод последовательных слияний')

plt.legend()

plt.grid(True)

plt.show()

print("Author - Moskalenko Vitaly Alexandrovich UIB-311")

# **ПРИЛОЖЕНИЕ Б**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

def Kmeans(matrica, k, Max=100, tol=1e-4):

Centroids=matrica[np.random.choice(matrica.shape[0], size=1, replace=False)]

for \_ in range(1, k):

Distants=np.linalg.norm(matrica[:, np.newaxis]-Centroids, axis=2)

MinDistants=np.min(Distants, axis=1)

NewCentroid=matrica[np.random.choice(matrica.shape[0], p=MinDistants/np.sum(MinDistants))]

Centroids=np.vstack([Centroids, NewCentroid])

for iteration in range(Max):

Distants=np.linalg.norm(matrica[:, np.newaxis]-Centroids, axis=2)

Labels=np.argmin(Distants, axis=1)

NewCentroids=np.array([matrica[Labels==i].mean(axis=0) for i in range(k)])

if np.all(np.abs(NewCentroids-Centroids)<tol):

break

Centroids=NewCentroids

return Labels, Centroids

dannie = """

2 52

3 44

4 30

5 34

6 54

8 46

9 26

10 36

10 58

11 50

13 28

14 42

15 56

16 34

17 48

24 24

25 12

26 18

27 8

27 28

29 24

30 16

30 30

32 4

32 22

33 12

34 26

35 6

36 20

37 12

20 88

21 96

22 80

23 90

23 102

25 106

26 78

26 88

26 100

27 72

29 92

30 82

30 104

32 86

32 96

7 70

4 72

7 74

10 74

6 78

9 78

11 80

5 82

7 84

13 86

10 88

8 90

12 92

9 96

11 100

"""

matrica=np.array([list(map(int, line.split())) for line in dannie.strip().split("\n")])

Mashtab=StandardScaler()

MatricaNormal=Mashtab.fit\_transform(matrica)

k=4

Labels, Centroids=Kmeans(MatricaNormal, k)

CentroidsOriginalMashtab=Mashtab.inverse\_transform(Centroids)

plt.figure(figsize=(10, 6))

Scatter=plt.scatter(matrica[:, 0], matrica[:, 1], c=Labels)

for i in range(k):

plt.scatter([], [], color=Scatter.cmap(Scatter.norm(i)), label=f'Кластер {i+1}')

plt.scatter(CentroidsOriginalMashtab[:, 0], CentroidsOriginalMashtab[:, 1], c='black', marker='X', s=200, label='Центроиды кластеров')

plt.title('Алгоритм k-средних')

plt.xlabel('Признак 3')

plt.ylabel('Признак 4')

plt.legend(loc='upper right')

plt.grid(True)

plt.show()

print("Author - Moskalenko Vitaly Alexandrovich UIB-311")

# **ПРИЛОЖЕНИЕ В**

Вариант: №3643810

Таблица 1 - ОБУЧАЮЩАЯ ВЫБОРКА

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Класс | Объект | ПРИЗНАКИ | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| А | 1 | 26 | 24 | 2 | 52 | 11 | 26 | 22 | 26 | 23 | 8 |
| 2 | 23 | 22 | 3 | 44 | 13 | 22 | 26 | 23 | 24 | 8 |
| 3 | 27 | 29 | 4 | 30 | 19 | 15 | 38 | 21 | 22 | 7 |
| 4 | 28 | 21 | 5 | 34 | 16 | 17 | 32 | 29 | 28 | 7 |
| 5 | 26 | 29 | 6 | 54 | 12 | 27 | 24 | 23 | 28 | 9 |
| 6 | 23 | 22 | 8 | 46 | 11 | 23 | 22 | 21 | 26 | 9 |
| 7 | 21 | 27 | 9 | 26 | 17 | 13 | 34 | 26 | 23 | 8 |
| 8 | 29 | 25 | 10 | 36 | 18 | 18 | 36 | 23 | 24 | 8 |
| 9 | 28 | 28 | 10 | 58 | 13 | 29 | 26 | 21 | 22 | 7 |
| 10 | 23 | 26 | 11 | 50 | 11 | 25 | 22 | 29 | 28 | 7 |
| 11 | 29 | 25 | 13 | 28 | 15 | 14 | 30 | 23 | 28 | 9 |
| 12 | 28 | 28 | 14 | 42 | 16 | 21 | 32 | 21 | 26 | 9 |
| 13 | 23 | 26 | 15 | 56 | 19 | 28 | 38 | 29 | 23 | 8 |
| 14 | 21 | 23 | 16 | 34 | 17 | 17 | 34 | 25 | 21 | 7 |
| 15 | 26 | 24 | 17 | 48 | 12 | 24 | 24 | 23 | 29 | 10 |
| В | 1 | 23 | 22 | 24 | 24 | 13 | 12 | 26 | 24 | 23 | 8 |
| 2 | 27 | 29 | 25 | 12 | 11 | 6 | 22 | 22 | 24 | 8 |
| 3 | 28 | 21 | 26 | 18 | 15 | 9 | 30 | 27 | 29 | 10 |
| 4 | 26 | 29 | 27 | 8 | 16 | 4 | 32 | 25 | 22 | 7 |
| 5 | 23 | 22 | 27 | 28 | 19 | 14 | 38 | 28 | 27 | 9 |
| 6 | 21 | 27 | 29 | 24 | 17 | 12 | 34 | 26 | 25 | 8 |
| 7 | 29 | 25 | 30 | 16 | 13 | 8 | 26 | 25 | 26 | 9 |
| 8 | 28 | 28 | 30 | 30 | 19 | 15 | 38 | 28 | 23 | 8 |
| 9 | 23 | 26 | 32 | 4 | 16 | 2 | 32 | 26 | 21 | 7 |
| 10 | 21 | 23 | 32 | 22 | 12 | 11 | 24 | 23 | 29 | 10 |
| 11 | 28 | 27 | 33 | 12 | 11 | 6 | 22 | 27 | 29 | 10 |
| 12 | 26 | 25 | 34 | 26 | 17 | 13 | 34 | 26 | 23 | 8 |
| 13 | 21 | 23 | 35 | 6 | 18 | 3 | 36 | 23 | 24 | 8 |
| 14 | 29 | 24 | 36 | 20 | 11 | 10 | 22 | 21 | 22 | 7 |
| 15 | 26 | 24 | 37 | 12 | 14 | 6 | 28 | 29 | 28 | 7 |
| С | 1 | 23 | 22 | 20 | 88 | 17 | 44 | 34 | 23 | 28 | 9 |
| 2 | 27 | 29 | 21 | 96 | 13 | 48 | 26 | 21 | 26 | 9 |
| 3 | 28 | 21 | 22 | 80 | 19 | 40 | 38 | 29 | 23 | 8 |
| 4 | 26 | 29 | 23 | 90 | 16 | 45 | 32 | 25 | 21 | 7 |
| 5 | 23 | 22 | 23 | 102 | 12 | 51 | 24 | 23 | 29 | 10 |
| 6 | 21 | 27 | 25 | 106 | 11 | 53 | 22 | 25 | 22 | 7 |
| 7 | 29 | 25 | 26 | 78 | 17 | 39 | 34 | 28 | 25 | 8 |
| 8 | 28 | 28 | 26 | 88 | 18 | 44 | 36 | 26 | 28 | 9 |
| 9 | 23 | 26 | 26 | 100 | 11 | 50 | 22 | 27 | 29 | 10 |
| 10 | 21 | 23 | 27 | 72 | 14 | 36 | 28 | 25 | 22 | 7 |
| 11 | 26 | 24 | 29 | 92 | 15 | 46 | 30 | 28 | 27 | 9 |
| 12 | 26 | 22 | 30 | 82 | 13 | 41 | 26 | 26 | 23 | 8 |
| 13 | 21 | 23 | 30 | 104 | 19 | 52 | 38 | 23 | 24 | 8 |
| 14 | 29 | 24 | 32 | 86 | 17 | 43 | 34 | 21 | 22 | 7 |
| 15 | 26 | 24 | 32 | 96 | 12 | 48 | 24 | 29 | 28 | 7 |
| D | 1 | 23 | 22 | 7 | 70 | 11 | 35 | 22 | 23 | 28 | 9 |
| 2 | 27 | 29 | 4 | 72 | 19 | 36 | 38 | 21 | 26 | 9 |
| 3 | 28 | 21 | 7 | 74 | 13 | 37 | 26 | 29 | 23 | 8 |
| 4 | 26 | 29 | 10 | 74 | 10 | 37 | 20 | 25 | 21 | 7 |
| 5 | 23 | 22 | 6 | 78 | 11 | 39 | 22 | 23 | 29 | 10 |
| 6 | 21 | 27 | 9 | 78 | 18 | 39 | 36 | 24 | 23 | 8 |
| 7 | 29 | 25 | 11 | 80 | 15 | 40 | 30 | 22 | 24 | 8 |
| 8 | 28 | 28 | 5 | 82 | 11 | 41 | 22 | 27 | 29 | 10 |
| 9 | 23 | 26 | 7 | 84 | 13 | 42 | 26 | 29 | 23 | 8 |
| 10 | 25 | 25 | 13 | 86 | 10 | 43 | 20 | 22 | 21 | 7 |
| 11 | 28 | 23 | 10 | 88 | 17 | 44 | 34 | 28 | 29 | 10 |
| 12 | 26 | 24 | 8 | 90 | 11 | 45 | 22 | 26 | 23 | 8 |
| 13 | 26 | 24 | 12 | 92 | 18 | 46 | 36 | 23 | 24 | 8 |

Продолжение таблицы 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 14 | 23 | 22 | 9 | 96 | 15 | 48 | 30 | 21 | 22 | 7 |
| 15 | 27 | 29 | 11 | 100 | 11 | 50 | 22 | 29 | 28 | 7 |