ЛЕКЦИЯ 1. ОБОСНОВАНИЕ ПРИМЕНЕНИЯ В ЭВМ СИСТЕМ СЧИСЛЕНИЯ С ОСНОВАНИЕМ 2. ФОРМУЛЫ ДЛЯ ГЕНЕРАЦИИ ПРОСТЫХ ЧИСЕЛ. АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ.

НАИБОЛЕЕ ЭФФЕКТИВНОЕ ОСНОВАНИЕ СИТЕМЫ СЧИСЛЕНИЯ С ТОЧКИ ЗРЕНИЯ РЕАЛИЗАЦИИ В ЭВМ

Предположим, что при разработке компьютера нового поколения встал вопрос о выборе системы счисления для представления в нем целых чисел. При этом будем считать, что имеется выбор аппаратной базы, которая позволяет использовать регистры с двумя состояниями (бинарные), с тремя состояниями, четырьмя и т.д. Какая же система счисления будет наиболее эффективна с точки зрения её стоимости?

Пусть стоимость системы с b состояниями пропорциональна b, так что схема с тремя состояниями на 50% дороже бинарной, а с четырьмя – в два раза дороже бинарной.

Предположим, что необходимо хранить в регистрах целые числа от 0 до некоторого максимально числа M. Представление целых чисел от 0 до M в системе счисления по основанию b требует $\approx \log_b(M+1)$ разрядов, например, для представления всех целых чисел от 0 до 999999 в десятичной системе требуется $\log_{10}(1000000) = 6$ цифр.

Таким образом, следует ожидать, что стоимость регистра с равна произведению количества требующихся цифр на стоимость представления каждой цифры:

$$c = k \log_b(M+1) * b,$$

где c – стоимость регистра, а k - константа, определяющая коэффициент стоимости. Таким образом, нужно найти такое значение b, которое бы минимизировало стоимость с для заданного M.

Найдя производную от указанной функции, имеем:

$$\frac{d}{db} c = k \ln(M+1) \frac{\ln b - 1}{\ln^2 b}.$$

Производная равна 0 при $\ln b = 1$, т.е. при $b = e$.

Данный результат говорит о том, что минимальную стоимость имела бы система счисления с основанием e=2.71828…, и наиболее эффективными системами является, таким образом системы с основаниями 2 и 3. При этом отношение стоимости регистра при использовании основания 2 к стоимости регистра при использовании основания 3 равно:

$$\frac{c(2)}{c(3)} = \frac{2\ln 3}{3\ln 2} \approx 1.056.$$

Таким образом, использование системы счисления по основанию 2 несколько дороже, чем использование системы по основанию 3, но на очень небольшую величину.

ФОРМУЛЫ ДЛЯ ПРОСТЫХ ЧИСЕЛ

Проблема генерации простых чисел является исключительно важной во многих практических задачах программирования. Особенно высокую значимость она имеет в криптозадачах (шифрование, расшифровывание и криптоанализ).

Известно несколько формул для простых чисел, которые по заданному n возвращают значение для n-го простого числа.

Первая формула Вилланса (C.P. Willans)

$$p_n = 1 + \sum_{m=1}^{2^n} \left\lfloor \sqrt[n]{n} \left(\sum_{x=1}^{m} \left\lfloor \cos^2 \pi \frac{(x-1)!+1}{x} \right\rfloor \right)^{-1/n} \right\rfloor,$$

где $\lfloor * \rfloor$ - операция взятия целой части числа.

Вторая формула Вилланса

$$p_n = \sum_{m=1}^{2^n} mF(m) \left\lfloor 2^{-|\pi(m)-n|} \right\rfloor,$$

где F и π - следующие функции:

$$F(x) = \left\lfloor \cos^2 \pi \frac{(x-1)!+1}{x} \right\rfloor = \begin{cases} 1, \text{ x - простое или } 1 \\ 0, \text{ x - составное} \end{cases}$$

$\pi(m) = -1 + \sum_{x=1}^{m} F(x)$ - количество простых чисел, не превышающих m.

Таким образом, $mF(m)$ равно m, если m - простое число или 1, и 0, если m - составное число.

Третья формула Вилланса

Третья формула Вилланса для простых чисел не содержит ни одной «неаналитической функции» типа модуля или взятия целой части.

$$p_n = 2 + \sum_{m=2}^{2^n} \sin \left(\frac{\pi}{2} \cdot 2^{\prod_{i=0}^{n-1} \left(\sum_{x=2}^{m} \frac{\sin^2 \pi \frac{((x-1)!)^2}{x}}{\sin^2 \frac{\pi}{x}} \right)} \right).$$

Четвертая формула Вилланса

Четвертая формула Вилланса выражает простое число p_{n+1} через p_n:

$$p_{n+1} = 1 + p_n + \sum_{i=1}^{2p_n} \prod_{j=1}^{i} f(p_n + j) \text{ , где}$$

$$f(x) = \left\lfloor \cos^2 \pi \frac{((x-1)!)^2}{x} \right\rfloor$$

Формула Вормелла (C.P. Wormell)

Формулы Вилланса были усовершенствованы Вормеллом, который «убрал» из них тригонометрические функции и функцию взятия целой части. Вычисления по формуле Вормелла в принципе могут быть проведены с помощью простой компьютерной программы с использованием только целочисленной арифметики.

$$p_n = \frac{3}{2} + 2^{n-1} - \frac{1}{2} \sum_{m=2}^{2^n} (-1)^{2^{s(m)}} \text{ , где}$$

$$s(m) = \prod_{r=1}^{n} \left(1 - r + \frac{m-1}{2} + \frac{1}{2} \sum_{x=2}^{m} (-1)^{2^{B(x)}} \right)^2 \text{ , где}$$

$$B(x) = \prod_{a=2}^{x} \prod_{b=2}^{x} (x - ab)^2$$

АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ

Пусть имеется множество произвольных объектов x_i. Кластеризация - это объединение объектов в группы (кластеры) на основе схожести признаков для объектов одной группы и отличий между группами. Большинство алгоритмов кластеризации не опираются на традиционные для статистических методов допущения; они могут использоваться в условиях почти полного отсутствия информации о законах распределения данных.

Задача кластеризации состоит в разбиении объектов из множества x_i на несколько подмножеств (кластеров), в которых объекты более схожи между собой, чем с объектами из других кластеров. В метрическом пространстве "схожесть" обычно определяют через расстояние. Расстояние может рассчитываться как между исходными объектами x_i, так и от этих объектов к объектам - прототипам кластеров. Обычно координаты прототипов заранее неизвестны - они находятся одновременно с разбиением данных на кластеры.

Существует множество методов кластеризации, которые можно классифицировать на четкие и нечеткие. Четкие методы кластеризации разбивают исходное множество объектов

на несколько непересекающихся подмножеств. При этом любой объект из x_i принадлежит только одному кластеру. Нечеткие методы кластеризации позволяют одному и тому же объекту принадлежать одновременно нескольким (или даже всем) кластерам, но с различной степенью. Нечеткая кластеризация во многих ситуациях более "естественна", чем четкая, например, для объектов, расположенных на границе кластеров.

Методы кластеризации также классифицируются по тому, определено ли количество кластеров заранее или нет. В последнем случае количество кластеров определяется в ходе выполнения алгоритма на основе распределения исходных данных.

В данной лекции рассматривается так называемый алгоритм горной кластеризации, который не требует задания количества кластеров. Метод предложен Р. Ягером и Д. Филевым в 1993 г.

Пусть $D(x_i, x_j)$ - функция, определяющая расстояние от объекта x_i до объекта x_j исходного множества объектов.

На первом шаге горной кластеризации определяют точки, которые могут быть центрами кластеров. Если о центрах кластеров заранее ничего не известно, то в качестве потенциальных центров удобно принять сами объекты. Обозначим множество потенциальных центров кластеров Z_h. Если центрами кластеров считаются сами объекты, то тогда $Z_h = x_k$.

На втором шаге для каждой такой точки рассчитывается значение потенциала, показывающего возможность формирования кластера в ее окрестности. Чем плотнее расположены объекты в окрестности потенциального центра кластера, тем выше значение его потенциала. После этого итерационно выбираются центры кластеров среди точек с максимальными потенциалами. Потенциал центров кластеров рассчитывается по следующей формуле:

$$P_1(Z_h) = \sum_{k=1}^{M} \exp(-\alpha D(Z_h, x_k)),$$

где M - количество объектов x_i, exp – функция экспоненты, α - положительная константа, характеризующая масштаб расстояний между объектами. В простейших случаях удобно полагать, что α равно единице, делённой на среднее расстояние между объектами.

В случае, когда объекты кластеризации заданы двумя признаками (точки на плоскости), графическое изображение распределения потенциала будет представлять собой поверхность, напоминающую горный рельеф. Отсюда и название - горный метод кластеризации.

На третьем шаге алгоритма в качестве центров кластеров выбирают координаты "горных" вершин. Для этого центром первого кластера V_1 назначают точку с наибольшим потенциалом. Обычно, наивысшая вершина окружена несколькими достаточно высокими пиками. Поэтому назначение центром следующего кластера точки с максимальным потен-

циалом среди оставшихся вершин привело бы к выделению большого числа близко расположенных центров кластеров. Чтобы выбрать следующий центр кластера необходимо вначале исключить влияние только что найденного кластера. Для этого значения потенциала для оставшихся возможных центров кластеров пересчитывается следующим образом: от текущих значений потенциала вычитают вклад центра только что найденного кластера. Перерасчет потенциала происходит по формуле:

$$P_2(Z_h) = P_1(Z_h) - P_1(V_1) \cdot \exp(-\beta \cdot D(Z_h, V_1)),$$

где β - положительная константа, характеризующая масштаб размера одного кластера (чем значение β меньше, тем больше размер кластера). В простейших случаях её можно принять равной α.

Центр второго кластера V_2 определяется как точка с максимальным значением потенциала $P_2(Z_h)$. Затем снова пересчитывается значение потенциалов:

$$P_3(Z_h) = P_2(Z_h) - P_2(V_2) \cdot \exp(-\beta \cdot D(Z_h, V_2)).$$

Итерационная процедура пересчета потенциалов и выделения центров кластеров продолжается до тех пор, пока максимальное значение потенциала превышает некоторый порог. В простейших случаях этот порог можно считать равным половине значения максимального потенциала $P_1(V_1)$.

Точка x_i считается принадлежащим кластеру V_k, если расстояние до него $D(x_i, V_k)$ меньше, чем расстояние до других кластеров. В этом случае все точки будут принадлежать каким-нибудь кластерам. Если же принять, что x_i считается принадлежащим кластеру V_k, если расстояние до него $D(x_i, V_k)$ меньше, чем некоторое пороговое значение, то могут остаться точки, не принадлежащие ни одному из найденных кластеров.

Ниже приводятся рисунки, иллюстрирующие основные идеи метода горной кластеризации.

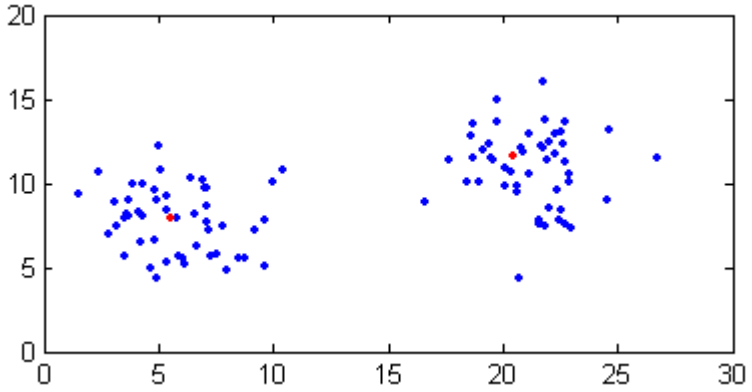

Рисунок 1. Точки на плоскости (синие) и найденные центры кластеров (красные)

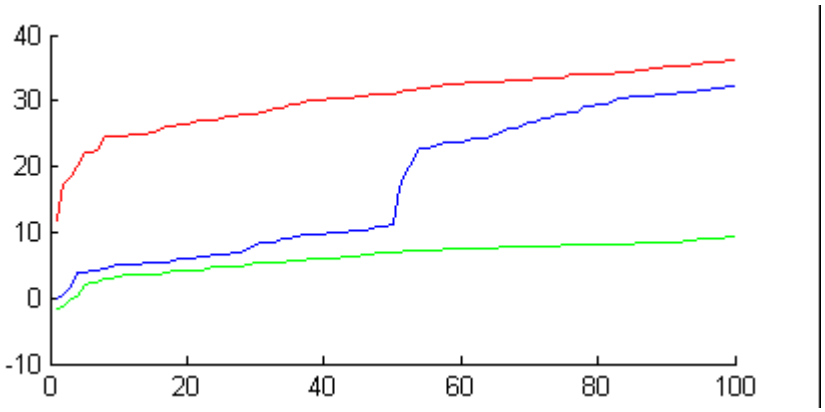

Рисунок 2. Упорядоченные значения потенциалов P1 (красная линия), P2 (синяя), P3 (зеленая).

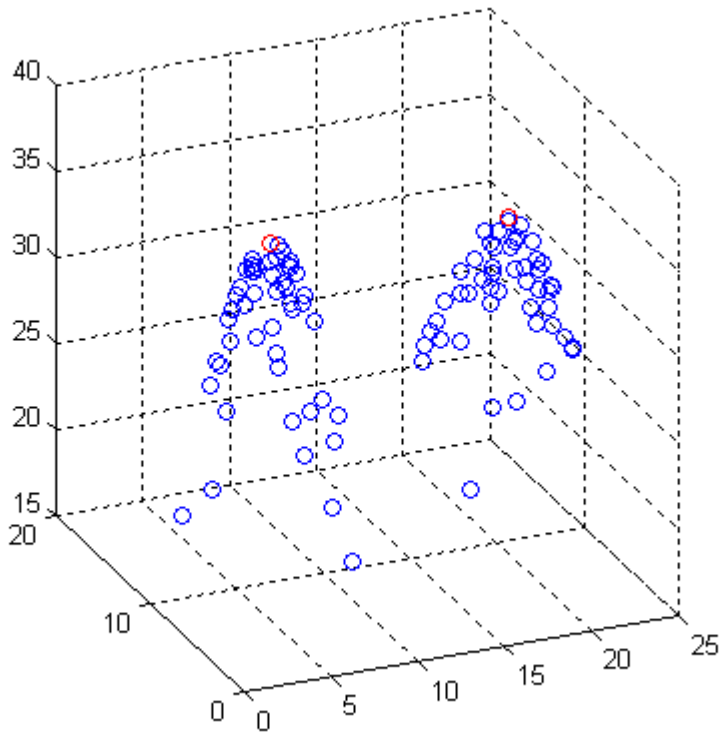

Рисунок 3. Значения потенциалов P1 точек (синим цветом) и значения потенциалов P1 центров кластеров (красным)

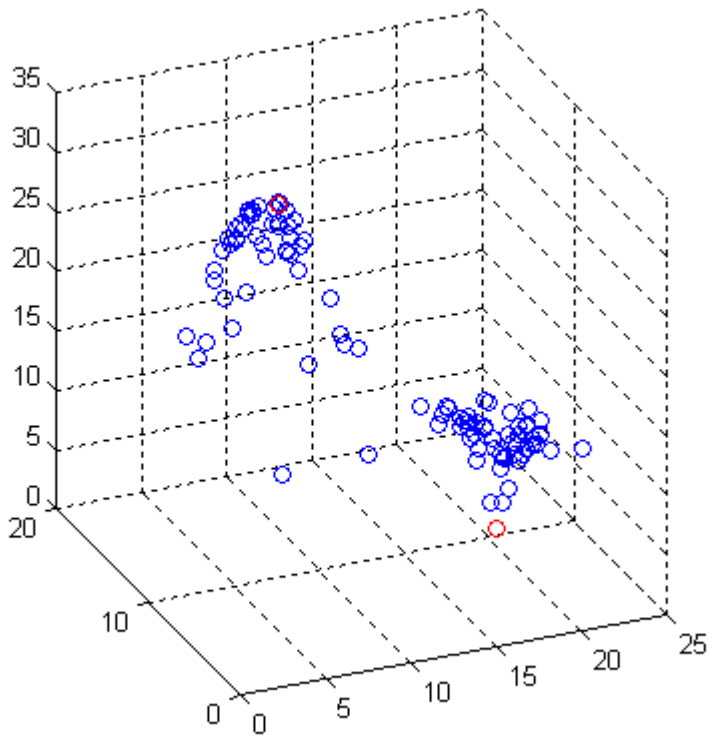

Рисунок 4. Значения потенциалов P2 точек (синим цветом) и значения потенциалов P2 центров кластеров (красным)

Задание к лекции 5

Реализовать алгоритм поиска в массиве целых чисел от 1 до заданного n простых чисел, называемый «Решето Эратосфена». Суть алгоритма состоит в том, что из массива чисел вначале «вычеркиваются» все, делящиеся на 2, затем делящиеся на 3, на 4, на 5 и т.д., вплоть до чисел, делящихся на n/2. Оставшиеся числа будут являться простыми.