Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по домашнему заданию № 1

по курсу «Анализ алгоритмов»

на тему: «Графовые представления»

Студент ИУ7-56Б (Группа) _____ (Подпись, дата) М. Ю. Вольняга (И. О. Фамилия)

Преподаватель _____ (Подпись, дата) Л. Л. Волкова (И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель данной лабораторной работы — описать четырьмя графовыми моделями (графом управления, информационным графом, операционной историей, информационной историей) последовательный алгоритм либо фрагмент алгоритма, содержащий от 15 значащих строк кода и от двух циклов, один из которых является вложенным в другой.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1) описать алгоритм кластеризации k-средних;

2) исследовать и разработать графовые модели для реализации алгоритма кластеризации k-средних;

3) провести анализ графовых моделей и обосновать возможность «распараллеливания» алгоритма кластеризации k-средних.

1 Аналитический раздел

В данном разделе будут рассмотрены графовые модели, алгоритмы классификации полнотекстовых документов, алгоритмы классификации без учителя и алгоритм k-средних.

1.1 Графовые модели

Графом системы управления называется граф $G = G(X, U)$, в котором множество вершин X интерпретирует множество элементов систем управления, а множество ребер U — множество связей между ними. Важным преимуществом модели в виде графа систем управления является возможность эффективного применения компьютерных технологий для автоматизации обнаружения критических структурных свойств исследуемой СУ [1].

Информационный граф — орграф информационных связей в программе или схеме программ. Необходимым условием наличия информационной связи между операндами операторов S_1 и S_2 является существование специального вида пути по управляющему графу от S_1 до S_2 — маршрута информационной связи, подтверждающего данную информационную связь [2].

Операционная история представляет собой последовательность преобразований, выполняемых при работе программы [3].

Если вершинам графа зависимостей соответствуют отдельные срабатывания операторов программы, то такой граф называется информационной историей выполнения программы. Информационная история содержит максимально подробную информацию о структуре информационных зависимостей анализируемой программы. Поэтому она используется при анализе программ с целью распараллеливания [4].

1.2 Алгоритмы классификации полнотекстовых документов

Классификация текстов — ключевая задача в компьютерной лингвистике, охватывающая алгоритмы с учителем и без учителя, и имеющая важное значение для обеспечения информационной безопасности. Алгоритмы с учителем используют предварительно размеченные данные для обучения, в то время как алгоритмы без учителя, такие как кластеризация, организуют

данные на основе внутренних закономерностей [5].

В данной лабораторной работе исследуется применение алгоритмов классификации без учителя для определения групп документов с помощью метода иерархической кластеризации с дивизимным подходом. Целью является выявление кластеров документов, таким образом, чтобы документы внутри одного кластера были максимально схожи по смыслу, а документы из различных кластеров — значительно отличались. Особенность данного подхода заключается в отсутствии необходимости в предварительной разметке данных и определении количества кластеров, что открывает широкие возможности для анализа неструктурированных данных [5]. Кластеризация — это разбиение элементов некоторого множества на группы по принципу схожести. Эти группы принято называть кластерами [6].

1.3 Алгоритмы классификации без учителя

Алгоритмы классификации без учителя разбивают набор документов на группы, где одна группа содержит родственные документы, а разные группы содержат разные документы. Без обучающего подмножества и известных категорий, алгоритм кластеризации автоматически определяет количество и состав кластеров, используя расстояния между документами [5].

Кластеризация текстов основана на идее, что похожие документы подходят к одним и тем же запросам, а разные документы подходят к разным запросам [5].

Исследование проводится над набором документов вида:

$$D = \{d_j \mid j = 1, \ldots, |D|\}, \tag{1.1}$$

содержащей разнообразные тематические классы. Цель алгоритмов классификации без учителя — автоматически классифицировать документы на кластеры вида:

$$C = \{c_j \mid j = 1, \ldots, |C|\}, \tag{1.2}$$

так чтобы каждый кластер представлял собой группу тематически схожих документов. Задача кластеризации сводится к определению оптимального множества кластеров C, удовлетворяющего заданным критериям качества [5].

1.4 Алгоритм k-средних

При заранее известном числе кластеров k, алгоритм k-средних начинает с некоторого начального разбиения документов и уточняет его, оптимизируя целевую функцию – среднеквадратичную ошибку кластеризации как среднеквадратичное расстояние между документами и центрами их кластеров:

$$e(D, C) = \sum_{j=1}^{k} \sum_{i:d_i \in C_j} \|d_i - \mu_j\|^2, \tag{1.3}$$

где μ_j — центр, или центроид, кластера C_j, $|C| = k$, вычисляющийся по формуле

$$\mu_j = \frac{1}{|C_j|} \sum_{i:d_i \in C_j} d_i, \tag{1.4}$$

где $|C_j|$ — количество документов в C_j. Идеальным кластером алгоритм k - средних считает сферу с центроидом в центре сферы.

Алгоритм k-средних состоит из следующих шагов [5].

1) *Вход:* множество проиндексированных документов D, количество кластеров k.

2) Назначить начальные центры для кластеров $\{\mu_j\},\ j = 1,\ ...,\ k$ случайным образом.

3) Установить каждому кластеру C_j пустой набор, $j = 1,\ ...,\ k$.

4) Для каждого документа $d_i \in D$ выполнить:

   – найти ближайший центр кластера $j^* := \arg\min_j \|\mu_j - d_i\|,\ j = 1,\ ...,\ k$;

   – добавить документ d_i в соответствующий кластер $C_{j^*} := C_{j^*} \cup \{d_i\}$.

5) Для каждого кластера C_j обновить центр как среднее его элементов:

$$\mu_j := \frac{1}{|C_j|} \sum_{i:d_i \in C_j} d_i.$$

6) Если условие остановки не достигнуто, вернуться к шагу 4.

7) *Выход:* множество центров кластеров $\{\mu_j\}$ и множество самих кластеров C.

Вывод

В данном разделе были рассмотрены графовые модели, алгоритмы классификации полнотекстовых документов, алгоритмы классификации без учителя и алгоритм k-средних.

2 Выполнение задания

В листинге 2.1 приведена реализация алгоритма k-средних.

Листинг 2.1 – Реализация алгоритма k-средних

```
MAXITER // -5
docs // -3
k    // -4
int n = docs.size // -1
std::vector<std::vector<double>> centroids(k,
   std::vector<double>(docs[0].size())); // -2
std::vector<std::vector<double>> kMeans(const
   std::vector<std::vector<double>> &docs, int n,
   std::vector<std::vector<double>> centroids, int k,) {
    // центроиды проинициализированы случ значениями
    size_t iter = 0; // 0
    std::vector<int> assignments(n, 0); // 1
    bool changed;                          // 2
    do {                                  // 3
        changed = false;                 // 4
        // Назначение точек кластерам
        for (int i = 0; i < n; ++i) {          // 5
            double bestDist = -1.0;           // 6
            int bestCluster = 0;             // 7
            for (int j = 0; j < k; ++j) { // 8
                double dist = cosineDistance(docs[i],
                   centroids[j]);     // 9
                if (dist > bestDist) {    // 10
                    bestDist = dist;     // 11
                    bestCluster = j;    // 12
                }
            }
            if (assignments[i] != bestCluster) { // 13
                assignments[i] = bestCluster;    // 14
                changed = true;                 // 15
            }
        }
        // Обновление центроидов
        centroids = updateCentroids(docs, assignments, k); // 16
    } while (changed && ++iter != MAXITER ); // 17
    return centroids;// 18
    }
```

2.1 Графовые представления

На рисунке 2.1 представлен граф управления. На рисунке 2.2 представлен информационный граф. На рисунке 2.3 представлена операционная история. На рисунке 2.4 представлена информационная история.

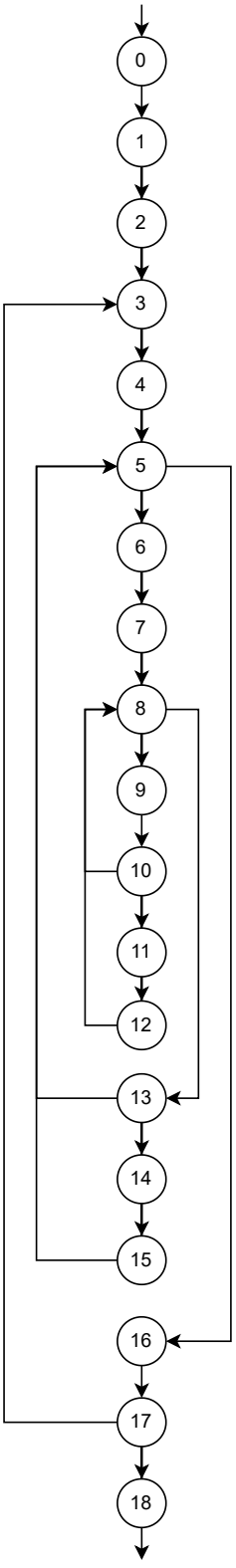

Рисунок 2.1 – Граф управления

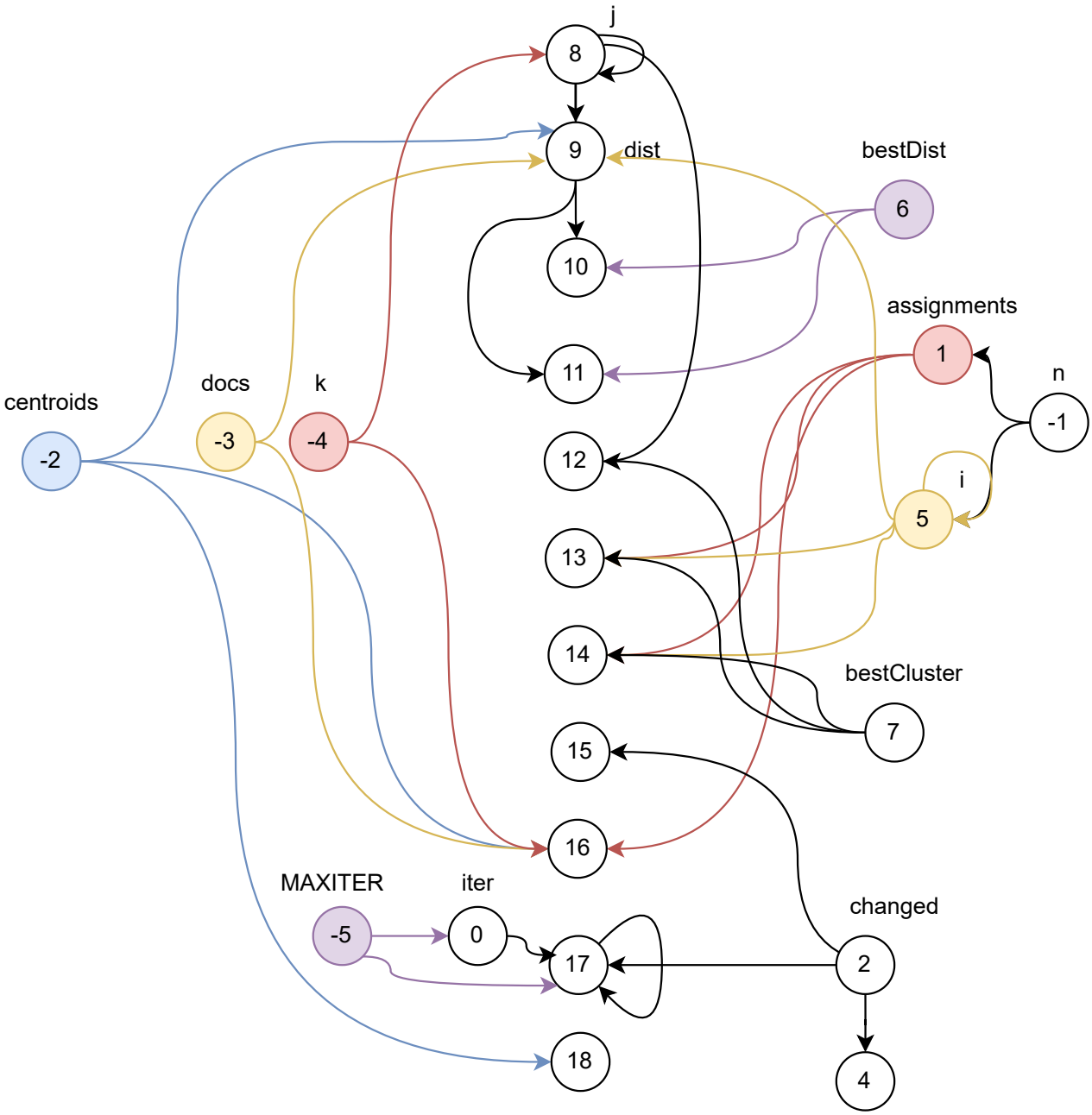

Рисунок 2.2 – Информационный граф

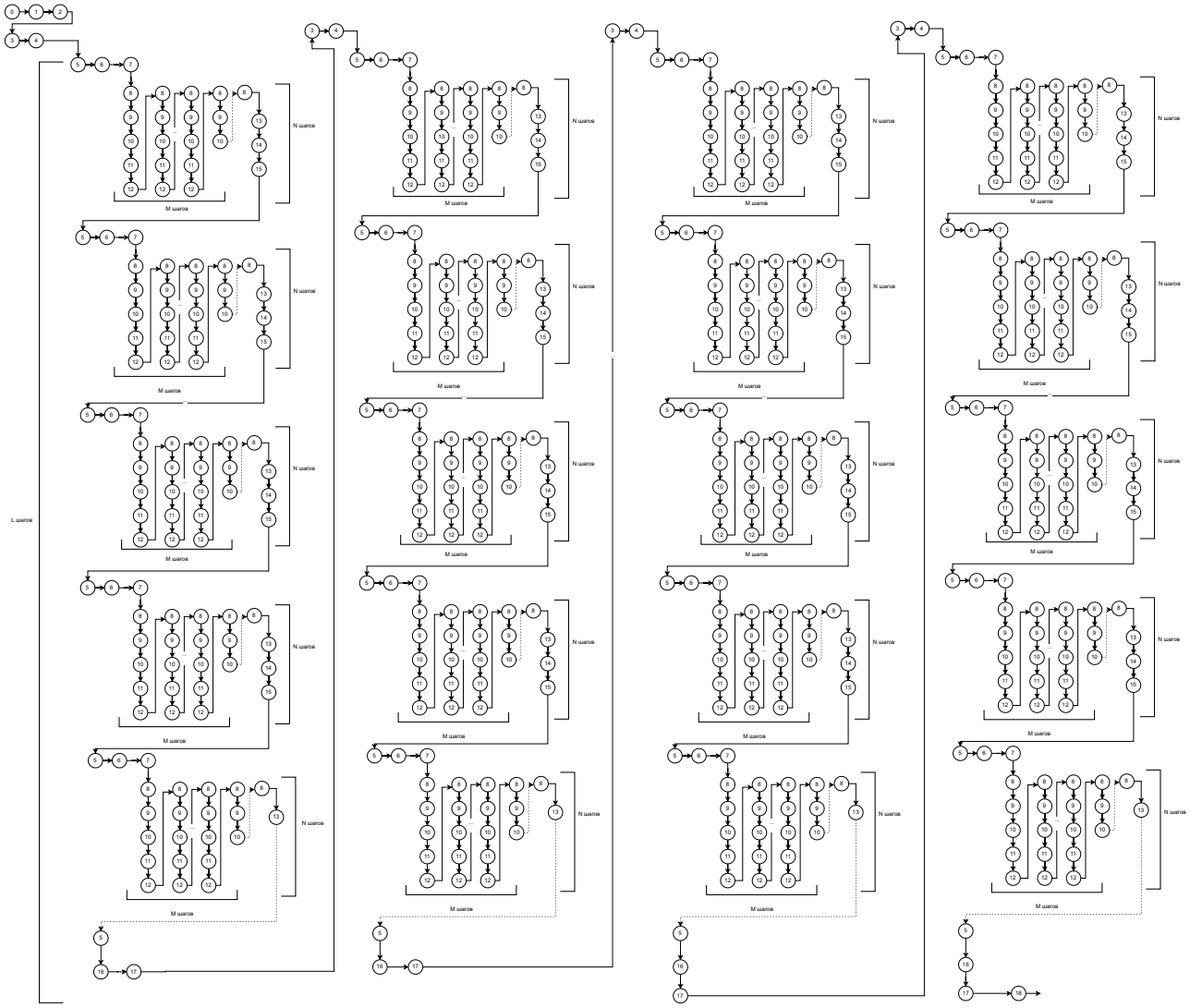

Рисунок 2.3 – Операционная история

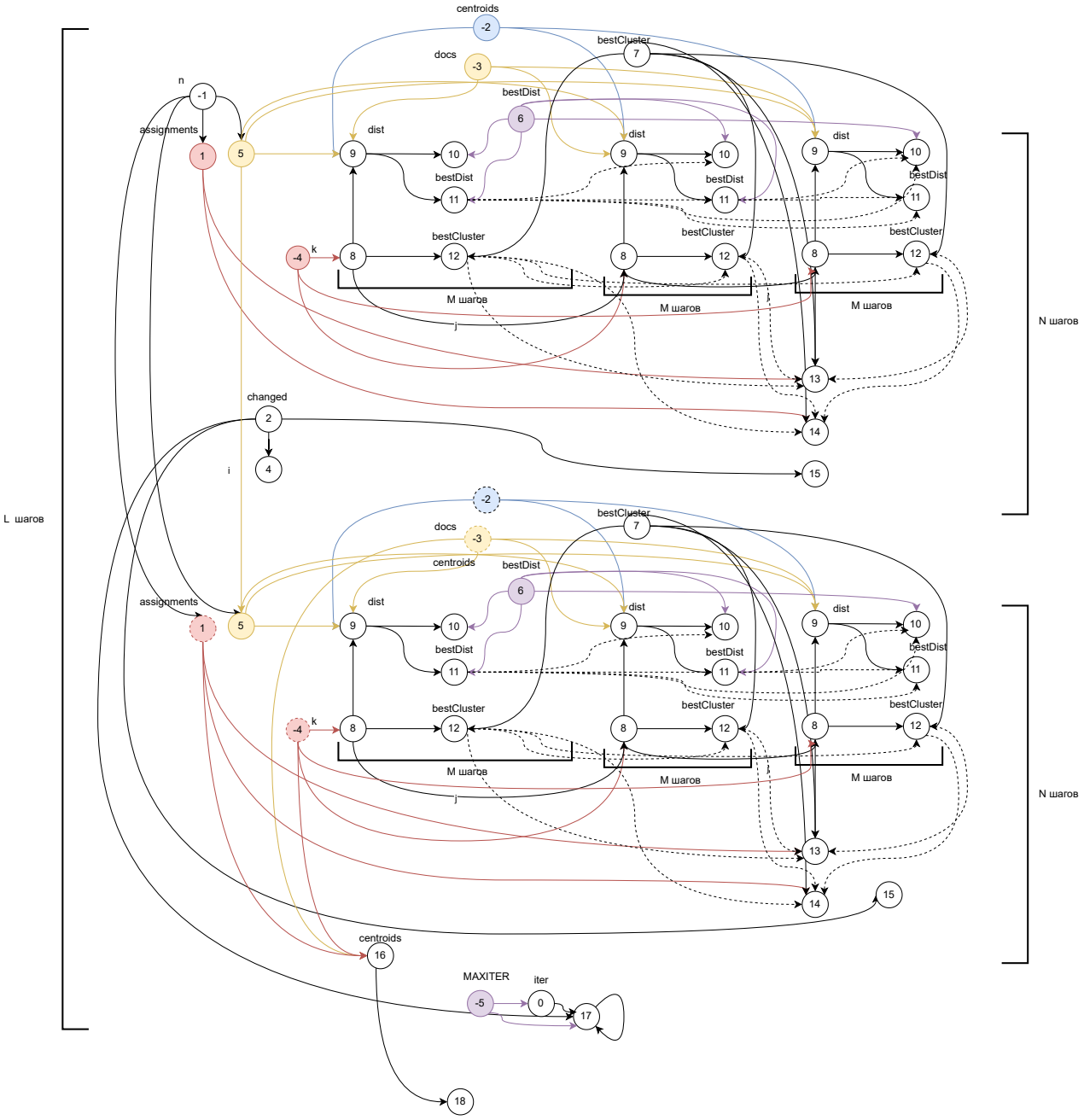

Рисунок 2.4 – Информационная история

2.2 Распараллеливание алгоритма

Проанализировав графы 2.1 – 2.4 можно сделать вывод, что распараллеливанию поддаются следующем этапы алгоритма:

– инициализация центроидов;

– назначение точек кластерам;

– обновление центроидов кластеров.

Вывод

В данном разделе была разработана реализация алгоритма k-средних, разработаны графовые модели, а также описаны возможные варианты распараллеливания алгоритма.

ЗАКЛЮЧЕНИЕ

Цель лабораторной работы достигнута, последовательный алгоритм классификации k-средних описан четырьмя графовыми моделями.

Для достижения поставленной цели были выполнены следующие задачи:

1) описан алгоритм кластеризации k-средних;
2) исследованы и разработаны графовые модели для реализации алгоритма кластеризации k-средних;
3) проведен анализ графовых моделей и обоснована возможность «распараллеливания» алгоритма кластеризации k-средних.

В результате анализа был получен вывод, что алгоритм можно распараллелить на следующих этапах:

- инициализация центроидов;
- назначение точек кластерам;
- обновление центроидов кластеров.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Представление структуры управления в виде графа [Электронный ресурс]. — Режим доступа: `https://kazedu.com/referat/98683/1` (дата обращения: 10.02.2024).
2. Информационный граф [Электронный ресурс]. — Режим доступа: `https://pco.iis.nsk.su/grapp` (дата обращения: 10.02.2024).
3. Свойства стандартных схем программ — Теория вычислительных процессов и структур [Электронный ресурс]. — Режим доступа: `https://studref.com/695998/informatika/svoystva_standartnyh_shem_programm` (дата обращения: 10.02.2024).
4. Технологии высокоскоростных вычислений [Электронный ресурс]. — Режим доступа: `https://libr.aues.kz/facultet/fit/is/32/umm/is_1.htm?ysclid=lqhzwtshkx353115151` (дата обращения: 10.02.2024).
5. *Большакова Е. И., Клышински Э. С., Ландэ Д. В., Носков А. А., Пескова О. В., Ягунова Е. В.* Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие. — МИЭМ, 2011. — С. 1—272.
6. *Котелина Н. О., Матвийчук Б. Р.* Кластеризация изображения методом К-средних // Вестник Сыктывкарского университета. — 2019. — Т. Выпуск 3 (32). — С. 102—106.