# Внешние спецификации программы

## Название и назначение программы

Программное средство «Lip4ikClusters» решает задачу кластеризации точек на плоскости с помощью технологии OpenMP с использованием метода k-средних.

## Команда сборки программы

Программное средство написано на языке программирования C++ с помощью технологии OpenMP. Код программы расположен в файле Lip4ikClusters.cpp, сборка исполняемого файла осуществляется следующей командой:

g++ -fopenmp <path>/Lip4ikClusters.cpp -o Lip4ikClusters -std=c++20 -O2 -march=native -mtune=native -floop-unroll-and-jam

Где path – путь к файлу Lip4ikClusters.cpp. Исполняемый файл будет создан в текущем каталоге.

## Формат вызова

Для запуска программы необходимо выполнить команду:

mpirun –np 1 <path>/Lip4ikClusters –n <n> -k <k> {-f <file> | -m <m>}

Где <path> – путь к файлу Lip4ikClusters.

Параметры:

* n – число OpenMP-потоков;
* k – число кластеров;
* file – путь к файлу с данными.
* m - количество точек

В случае если задан аргумент file, программа будет выполнятся в тестовом режиме, в случае когда задан аргумент m - в эксперементальном и данные сгенерируются автоматически.

## Входные данные

В режиме тестирования на вход программы передается файл, который содержит информацию о кластеризуемых точках в двумерном пространстве. Файл должен иметь формат ‘json’, соответствовать спецификации json и представлять собой текстовый файл следующего вида.

Тестовый файл должен содержать массив векторов следующего вида [x,y].

Где

* x – координата по оси абсцисс
* y – координата по оси ординат

Пример входного файла для двух точек с координатами x1=1, y1=2, x2=3, y2=4:

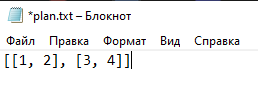


Рис 1 – Пример входного файла

## Выходные данные

При выполнении программы в тестовом режиме в файл output.json будет выведена информация о распределении точек по кластерам, выходной файл будет соответствовать формату json. Файл будет располагаться в каталоге, из которого производился запуск программы. Структура данных в файле представляет собой список словарей. Каждый словарь хранит информацию о кластере ключами являются поля “points” и “coordinate”. Значение по ключу “points” будет хранить список координат точек, отнесенных к кластеру, “coordinate” – координаты центроида.

Пример вывода программы для двух кластеров, в каждом из которых по две точки:

'[

{

"coordinate": [

0,

0

],

"points": [

[

1,

1

],

[

-1,

-1

]

]

},

{

"coordinate": [

10,

10

],

"points": [

[

9,

9

],

[

11,

11

]

]

}

]'

В приведенном примере первый центроид имеет координаты x=0, y=0, входные точки x1=1, y1=1 и x2=-1, y2=-1. Второй центроид координаты x=10, y=10 входные точки x1=9, y1=9 и x2=11, y2=11.

## Внешние эффекты

Сообщения об ошибках будут выведены в стандартный поток ошибок, других внешних эффектов программа не предусматривает.

Таблица 1 – Сообщения об ошибках

|  |  |
| --- | --- |
| Сообщение | Возникшая ошибка |
| Input parameters error | Неверное количество входных параметров или отсутствуют обязательные параметры. |
| File error <filename> : <message> | Ошибка при открытии файла <filename>. <message> - описание ошибки. |
| Incorrect input data: <message> | Неверно сформирован входной файл. Подробное описание ошибки формата описано в сообщении message.  Пример сообщения message:  parse error at line 3, column 8: syntax error while parsing array - unexpected '}'; expected ']' |