

## Техническое задание

На разработку алгоритма и программы для параллельной обработки графов

### 1. Исходные данные

- a. Исходные данные представлены в текстовом формате в виде координат вершин и дополнительных данных. Например:

	Area	XM	YM	Circ.	AR	Round	Solidity
1	4	49.000	10.500	0.740	4.000	0.250	1.000
2	5	159.100	11.700	1.000	1.553	0.644	0.909
3	4	345.500	11.000	1.000	2.031	0.492	0.800
4	4	570.000	10.500	0.740	4.000	0.250	1.000
5	4	586.000	11.000	1.000	1.000	1.000	1.000

где интересующие нас данные это XM, YM, а также имя вершины – первый столбец с цифрами.

- b. Исходные данные, являются результатом обработки одного изображения и хранятся в отдельных файлах в соответствии с именем исходного файла, которое (имя файла) необходимо сохранять для последующего использования в выходных данных.

### 2. Требования к алгоритму

- a. Алгоритм должен выделять группы точек по координатам XM, YM в отдельные графы.
- b. Условием выделения вершин в один граф является расстояние между вершинами, заданное в диапазоне значений от  $R_{\min}$  до  $R_{\max}$ .
- c. Графы должны быть оформлены в виде объектов соответствующего класса с указанным набором свойств и методов.
- d. По результатам обработки все графы, выделенные из одного файла, записываются в формате XML в один файл с исходным именем с добавлением расширения xml. Например:

```
<?xml version="1.0" encoding="UTF-8"?>
<ArrayOfSimplex xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://
- <Simplex Name="1" Radius="1" Diameter="2" Length="2" E="-1,6509215100492">
  - <Vertex Name="5" IndexInSimplex="0" Ec="2" x="274,3" y="10,9">
    <Neighbors ListOfIndex="{1}" ListOfNames="{6}" />
  </Vertex>
  - <Vertex Name="6" IndexInSimplex="1" Ec="1" x="284,5" y="10,5">
    <Neighbors ListOfIndex="{0;2}" ListOfNames="{5;40}" />
  </Vertex>
  - <Vertex Name="40" IndexInSimplex="2" Ec="2" x="286" y="19,5">
    <Neighbors ListOfIndex="{1}" ListOfNames="{6}" />
  </Vertex>
</Simplex>
- <Simplex Name="2" Radius="1" Diameter="2" Length="2" E="-1,80779107548303">
  - <Vertex Name="20" IndexInSimplex="0" Ec="2" x="950" y="11,3">
    <Neighbors ListOfIndex="{1}" ListOfNames="{29}" />
  </Vertex>
  - <Vertex Name="29" IndexInSimplex="1" Ec="1" x="940,5" y="13,5">
    <Neighbors ListOfIndex="{0;2}" ListOfNames="{20;44}" />
  </Vertex>
  - <Vertex Name="44" IndexInSimplex="2" Ec="2" x="939,1" y="22,7">
    <Neighbors ListOfIndex="{1}" ListOfNames="{29}" />
  </Vertex>
</Simplex>
- <Simplex Name="3" Radius="1" Diameter="1" Length="1" E="-0,936790567751564">
  - <Vertex Name="31" IndexInSimplex="0" Ec="1" x="162,5" y="15">
    <Neighbors ListOfIndex="{1}" ListOfNames="{41}" />
  </Vertex>
  - <Vertex Name="41" IndexInSimplex="1" Ec="1" x="170,833" y="19,833">
    <Neighbors ListOfIndex="{0}" ListOfNames="{31}" />
  </Vertex>
</Simplex>
```

- e. Алгоритм должен выполнять параллельную или высокопроизводительную обработку данных с использованием потоков/ядер ЦПУ или ГПУ.

3. Требования к классу «граф/graph».

- a. Класс должен иметь имя – порядковый номер, уникальный в пределах одного обрабатываемого файла данных – тип строка.
- b. Свойство энергия графа/Energy – тип реальное число двойной точности.
- c. Свойство список вершин – объектов класса Вершина/Vertex с соответствующими свойствами – тип список вершин.
- d. Методы:
  - i. Получить число вершин в графе
  - ii. Получить объект вершина по номеру вершины в графе
  - iii. Получить/вычислить радиус графа
  - iv. Получить/вычислить диаметр графа
  - v. Получить/вычислить длину графа
  - vi. Получить список вершин в текстовом формате
  - vii. Получить/вычислить энергию графа для заданного  $R_0$
  - viii. Добавить объект вершина в список вершин графа
  - ix. Стандартный конструктор и деструктор

4. Требования к классу вершина/vertex

- a. Свойства:
  - i. Имя вершины – номер вершины из исходного файла – тип текст/строка
  - ii. Номер в графе – порядковый номер вершины в графе – тип целое
  - iii. Координата X – координата вершины из исходного файла – реальное двойной точности
  - iv. Координата Y – координата вершины из исходного файла – реальное двойной точности
  - v. Список соседей вершины – список объектов класса сосед/Neighbor
- b. Методы:
  - i. Получить число соседей
  - ii. Добавить соседа
  - iii. Получить объект сосед по номеру в списке
  - iv. Стандартный конструктор и деструктор

5. Требования к классу сосед/Neighbor

- a. Свойства:
  - i. Имя – номер вершины из исходного файла – тип текст/строка
  - ii. Номер в графе - порядковый номер вершины в графе – тип целое
- b. Стандартный конструктор и деструктор