Кулик Павел, ИС-241. Генерация случайного дерева. Нечётный вариант.

Постановка задачи

Требуется написать программу, реализующую алгоритм генерации случайного дерева с заданным количеством вершин и заданным ограничением на максимальную степень вершины. Программа должна также иметь функционал дополнения дерева до графа без кратных рёбер.

Реализация

Алгоритм, реализованный в программе, основывается на алгоритме GTD, описанном в статье “On Generating Random Network Structures: Trees”, Alexey S. Rodionov, Hyunseung Choo. Исходный код программы доступен на github.

Программа принимает на вход 3 аргумента:

- Строка: “tree” или “graph”;
- Число: количество вершин;
- Число: максимальная степень вершины.

После выполнения программа выводит в терминал текст скрипта, который добавляет вершины и рёбра сгенерированного графа в базу данных neo4j. Для запуска базы данных используется команда

```
docker compose up -d
```

Процесс генерации графа и добавления его в базу данных автоматизирован с помощью скрипта run_app.sh, принимающего на вход аргументы, аналогичные основной программе. После выполнения скрипта графическое отображение графа можно увидеть перейдя на http://localhost:7474/browser/.

Генерация нового графа удаляет старый граф из базы данных.

На последующих страницах представлены изображения нескольких деревьев и графов, сгенерированных с помощью программы. Направления рёбер на рисунках присутствуют только потому, что neo4j не поддерживает добавление рёбер без направления. Несмотря на отсутствие необходимости в направлениях рёбер, по ним можно найти корень – в него не направлено ни одного ребра.

Демонстрация

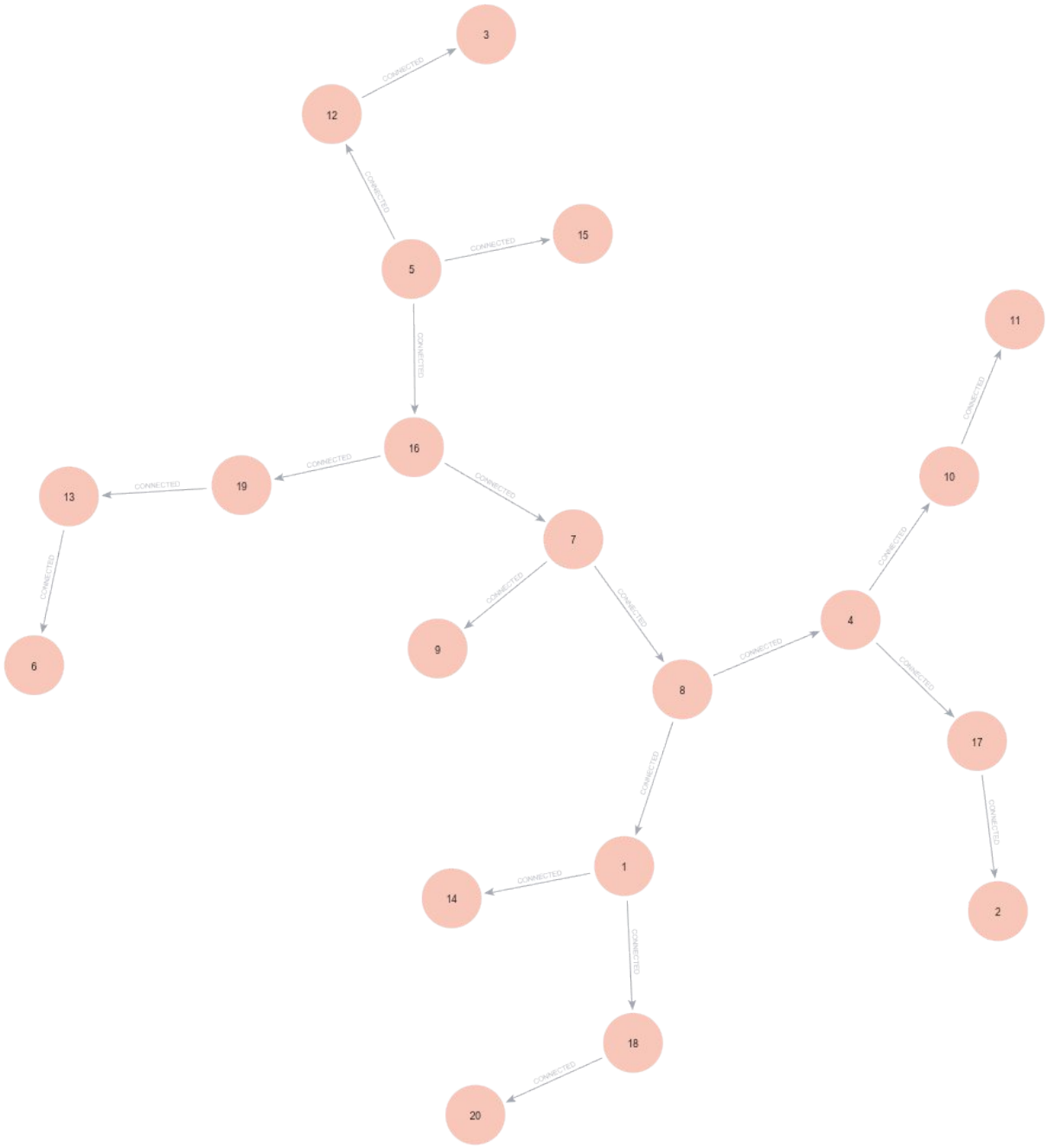

Рисунок 1 – дерево, 20 вершин, степень 3

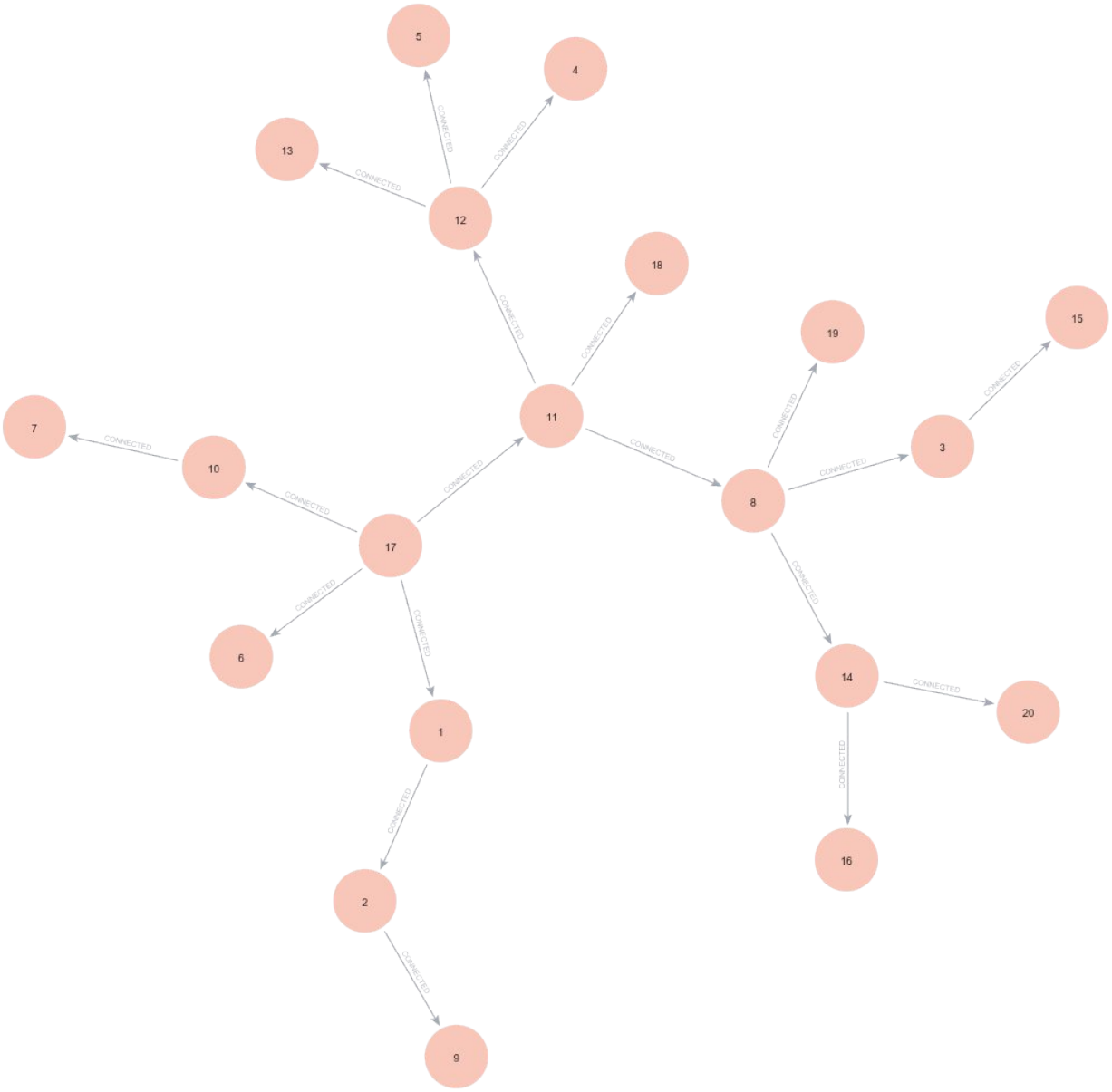

Рисунок 2 – дерево, 20 вершин, степень 4

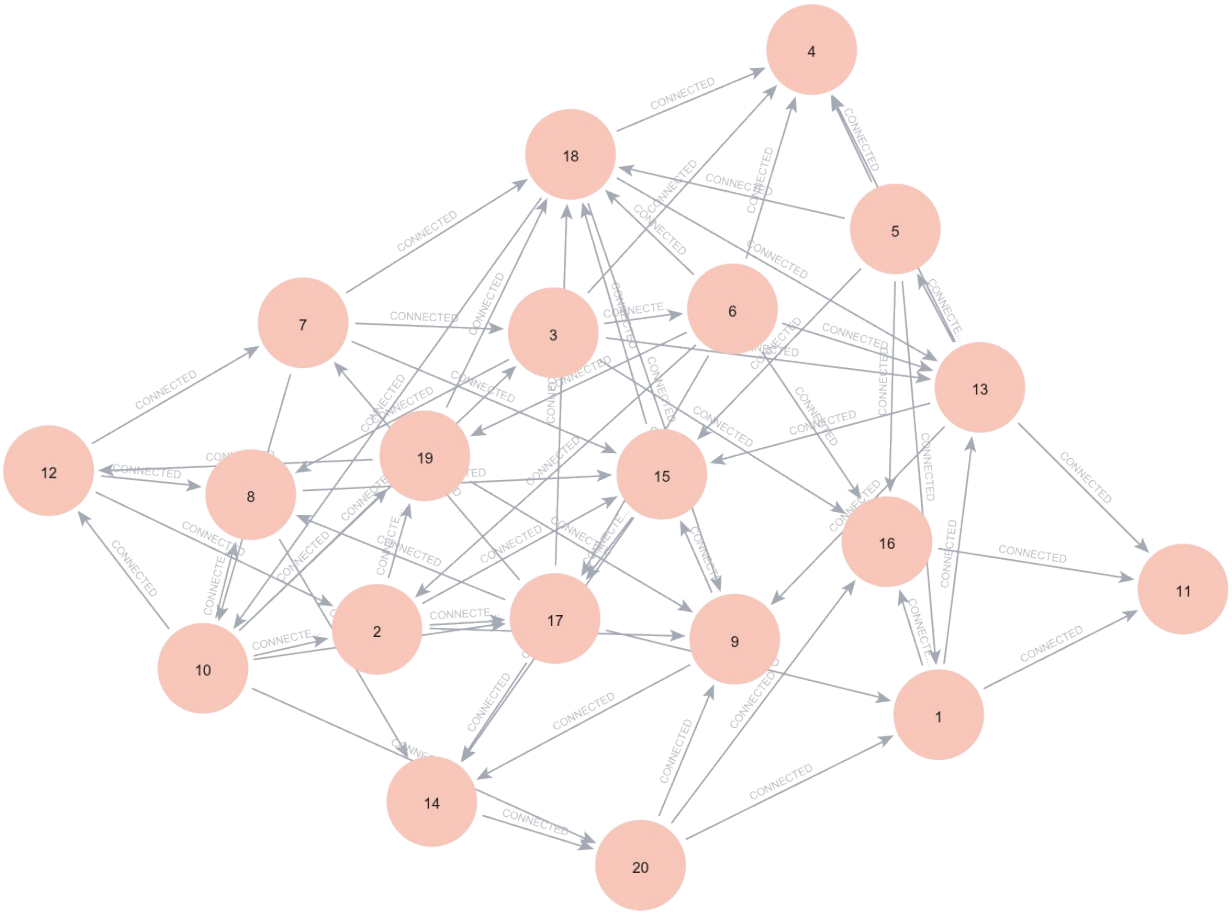

Рисунок 3 – граф, 20 вершин, 67 рёбер

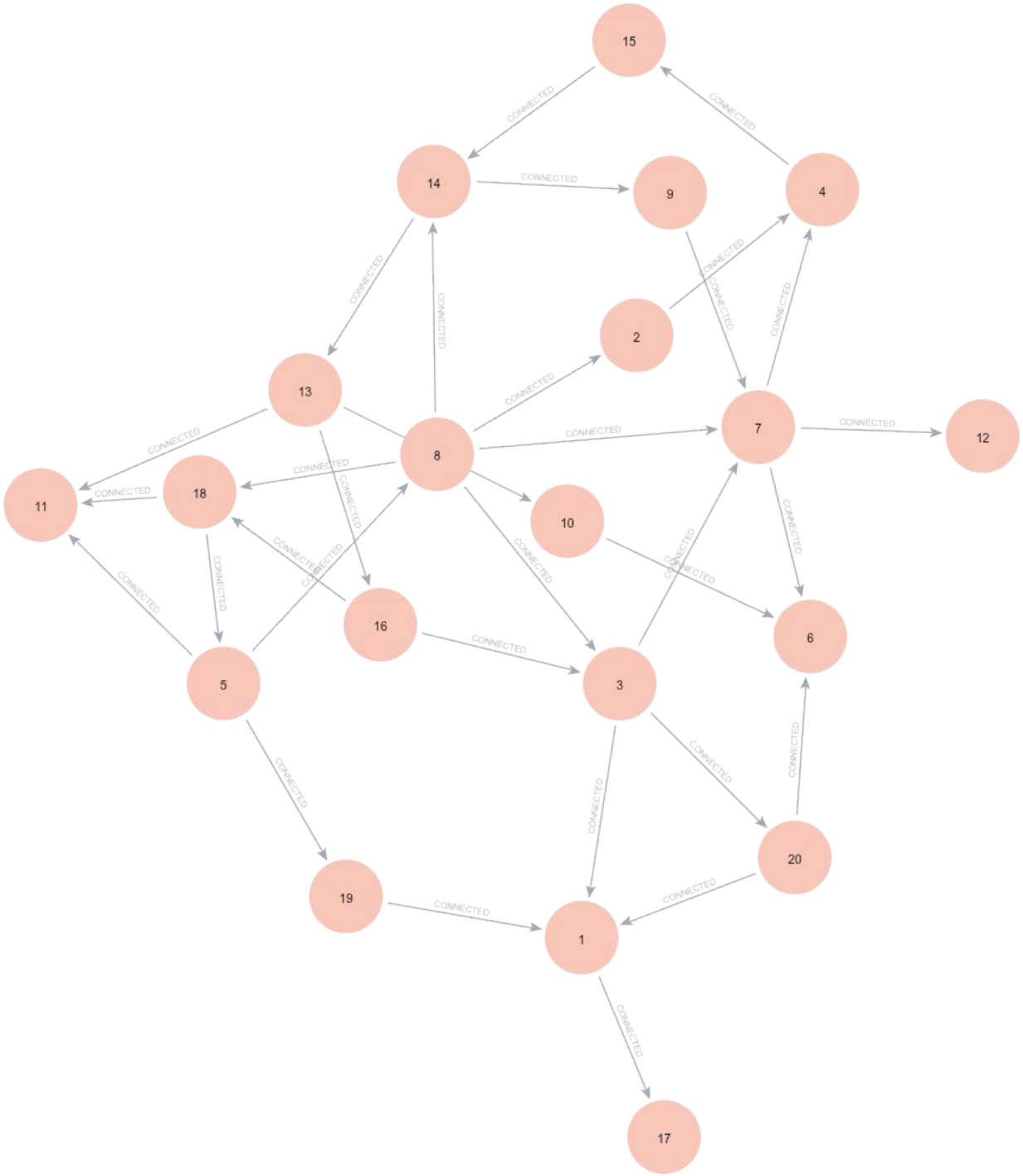

Рисунок 4 – граф, 20 вершин, 32 ребра