Inställningar

Computer Science for Mathematicians - Spring 19

Lab 6: Graph diameters

In this assignment, you will implement your own datastructure for undirected graphs and breadth-first search (BFS). This code will then be utilised in a small investigation of the diameter of random graphs.

1. A graph class

Implement the class `Graph` using an established graph representation you prefer.
The `Graph` class should have a reasonable interface hiding the implementation details. Two constructors are required:

1. `Graph()` shoud yield an empty graph.
2. `Graph(n,p)` is constructor that takes two arguments and returns a random graph. The arguments are the number of vertices in the graph and the probability of creating an edge between any two vertices.

One `Graph` method is required:

- `distance(start)` takes one argument, a start vertex, and computes the distance to all other vertices, returned in a suitable datastructure. The distance to $v \in V$ is computed as the minimum number of edges in a path from start to v. If there is no path to v, then that distance is reported as `-1`. The distances must be computed using BFS.

For grading

1. You should be able to discuss the time complexity of your BFS implementation. Is it $O(|V|+|E|)$ as it should be?
2. Is your class interface hiding the implementation details?

2. A module for reading graphs in DOT format

An old format for sharing graph data is called DOT (Wikipedia) and there are several Unix tools for visualising such graph files.
You will write a class called `GraphDotIO` that has a method `read_dot` that understands a subset of the DOT format. Although DOT is fairly simple, there are several details that we do not need in this lab. The list of example files below defines the subset of DOT. In words: your code must parse undirected graphs without weights and labels. For example:

```
graph Small {
  a -- b;
  b -- c;
}
```

is the graph with vertex set *V*={a, b, c} and edge set *E*={(a, b), (b, c)}.

Requirements for `read_dot`

- The method `read_dot` will take a file as parameter and return a graph object. Note that by reading from a file, your function will be able to parse data from `stdin` (System.in) as well as from files.
- An exception should be "thrown" if there is a problem with the input.
- You must construct at least three additional test files and argue for why they are meaningful to work with when testing your classes.

Note that it is *not* required that you recursive descent for parsing.

For grading

3. Show your testfiles and argue why they are useful.
4. Demonstrate that `read_dot` works on your testfiles and the example files (below).

3. Diameter investigation on random graphs

Write a Java program that investigates the mean diameter on *connected* random graphs, as implemented in `Graph.random(n,p)`. This class of random graphs is not guaranteed to be connected, but we can get around that by simply rejecting disconnected graphs when they are found. (*Note*: you do not need to explicitly implement such a test -- it is sufficient to discover this when computing the diameter.)
Your random graphs must have at least 100 vertices. The edge probability *p* should be varied between 0.1 and 0.9 in steps of 0.1. For each *p*, the mean diameter must be computed over at least 100 connected graphs.

The program should produce a simple result table to `stdout`.

For grading

5. What is the time complexity of your algorithm for computing the diameter of a graph?
6. You must be able to explain your code and how it works.
7. Show a result table for different values of *p*.

Example data

You can download sample data from here:

- line.dot
- circular.dot

Senast modifierad: fredag, 3 maj 2019, 13:51

◄ f9 — Syntaxanalys och rekursiv medåkning

Hoppa till... ▾

Inställningar

NAVIGATION

Hem
- Mitt hem
  - Mina kurser
    - Vetenskaplighet VT19
    - MT5002 VT19
    - MT5009 VT19
    - MT5012 VT19
    - MT6001 VT19
      - DA3018 VT19
        - Deltagare
        - Märken
        - Kompetenser
        - Betyg
          - 13 maj — 17 maj
            - **Lab 6: Graph diameters**
    - Kandidatprogram i matematik
    - MM2001 - ht16
    - Anmälning etenta Matematik I - ht16
    - Handledning - ht16
    - Mer...