

Due: 11:59pm, Thursday, Nov. 30, 2023

Learning Objectives

The goal of this assignment is to gain experience with graph traversals.

Instructions

Download the source files `graph.py` and `assignment4.py`.

Inside `graph.py` is a version of the `Graph` class, and `Vertex` and `Edge` classes, provided with the textbook.

The source file `assignment4.py` contains some starter code for question 1.

Question 1: (80 pts)

High degree vertices are often important in real-world contexts because they have many connections. However, for very large graphs, it may be difficult to find these vertices if, for example, the graph is too large to fit into memory. One option is to traverse the graph vertex by vertex to search for high degree vertices.

Consider the algorithm below.

```
1  ALGORITHM find_high_degree(graph G, vertex v)
2      set max = v
3      set n = neighbours of v
4      while n is not empty
5          set v = the highest degree vertex in n
6          if degree(v) > degree(max) then
7              set max = v
8          endif
9          set n = neighbours of v that remain unvisited
10     end
11     return max
```

Starting from an initial vertex, the algorithm traverses from one vertex to a neighbouring vertex, preferentially choosing the neighbour that has the highest degree. It does not visit a vertex that has been previously visited. (This means that it cannot go backward, for instance.) Once there are no further neighbouring vertices that are valid moves, the function returns the vertex it found that had the highest degree.

Implement the algorithm `find_high_degree(graph, vertex)`. This is a standalone function that accepts a graph and vertex, which are objects of the `Graph` and `Vertex` classes in the supplied `graph.py` source file. Note that this function is not a method of the `Graph` class. The code in `assignment4.py` can be used as a starting point.

Do not modify the `Graph` class given in the `graph.py` source file. (It does not contain all the methods discussed in the lectures, but does contain everything needed for this question.)

Question 2: (20 pts)

Draw an example of a single-component graph where algorithm `find_high_degree()` would fail to find the vertex with the highest degree. Indicate a starting vertex that would cause this failure.

Submission

Submit a single zip file containing the Python source file for Question 1 and the graph drawn for Question 2. Submit your drawing as a pdf, jpg or png file.

Python source code should be `*.py` plain text. The only file types allowed aside from Python source code (`*.py`) are pdfs (or png or jpg for question 2) and plain text (`*.txt`). Do not submit Word documents or rich text format documents. They will not be marked. Only submit a single zip archive. Do not submit files archived in rar format. That may result in your assignment not being graded.

Name all files with the format “`firstname_lastname_studentid_...`”. Make sure to include your name and student ID as comments at the top of all Python source files.

Late submissions will be subject to a 10% penalty for each hour past the deadline.

Attribution

Submissions must represent your independent work.

If your submitted work includes unacknowledged collaboration, code materials, ideas or other elements that are not your original work, it may be considered plagiarism or some other form of cheating under MUN general regulations 6.12.4.2 and academic penalties will be applied accordingly.

The submission of work that has been created by generative artificial intelligence (GAI) tools and presented as a student’s original work is considered an academic offence in this course. Using AI tools without proper citation constitutes plagiarism, and your work will be subject to the appropriate Memorial’s Academic Misconduct policy.

If your submission contains any contribution by others, including internet sources and classmates,

then you should include an attribution section detailing the extent of these contributions. This will also help distinguish what elements of the submission are original. You may not receive full credit if your original elements are insufficient, but can lessen penalties for plagiarism or copying if you acknowledge your sources.

Github

I encourage you to store and version your work on Github. It is good practice to do so as everyone uses git in the real world.

However, **it is a requirement that git repositories containing assignment material be private.** University regulations section 6.12.4.2 consider it cheating if you allow your work to be copied. There will be zero tolerance for this.