

CMSC 420 Fall 2024: Coding Project 6

Kruskal's Algorithm

with a

Disjoint-Set Data Structure

1 Due Date and Time

Due to Gradescope by Sunday 8 December at 11:59pm. You can submit as many times as you wish before that.

2 Get Your Hands Dirty!

This document is intentionally brief and much of what is written here will be more clear once you start looking at the provided files and submitting.

3 Assignment

We have provided the template `graph.py` which you will need to complete. More specifically you will fill in the code details to manage graphs and disjoint set data structures. More details are given below.

4 Details

Code that needs to be filled in:

- `def __init__(self, adjmat):`
Fill in the code which fills in `self.edgelist` from `self.adjmat`. Note that the code to fill in `self.adjmat` is already done as part of `test_graph.py` and `graph.py`.
- `def findrep(self, i) -> int:`
Find and return the component representative for the edge with index `i`, using path compression.
- `def union(self, i, j):`
Take the weighted union of the components with indices `i` and `j`.
- `def kruskal(self):`
Apply Kruskal's Algorithm to find the edges for a minimum spanning tree. Print a list of these edges in the order in which they were added.
- `def unkruskal(self):`
Apply Kruskal's Algorithm to find the edges for a minimum spanning tree. Print a list of the edges which were excluded (because including them would form a cycle) in the order in which they were excluded.

5 Additional Functions

You probably don't need any additional functions - please ask if you think you do!

6 What to Submit

You should only submit your completed `graph.py` code to Gradescope for grading. We suggest that you begin by uploading it as-is (it will run!), before you make any changes, just to see how the autograder works and what the tests look like. Please submit this file as soon as possible.

7 Testing

This is tested via the construction and processing of tracefiles.

- The first line in the tracefile indicates the number of vertices in the graph.
- The next lines make up the adjacency matrix for the graph.
- The final line is either `dump_edgelist`, which dumps the list of edges, `dump_adjmat`, which dumps the adjacency matrix, `kruskal`, which runs Kruskal's Algorithm and dumps a list of edges which are included in the order in which they are included, or `unkruskal`, which runs Kruskal's Algorithm and dumps a list of edges which are excluded in the order in which they are excluded, because inclusion would form a cycle.

You can see some examples by submitting the `graph.py` file as-is.

8 Local Testing

We have provided the testing file `test_graph.py` which you can use to test your code locally. Simply put the lines from a tracefile (either from the autograder or just make one up) into a file `whatever` and then run:

```
python3 test_graph.py -tf whatever
```