

## 3.9

Andrew Lee

March 10, 2017

For each node  $u$  in an undirected graph, let  $\text{twodegree}[u]$  be the sum of the degrees of  $u$ 's neighbors. Show how to compute the entire array of  $\text{twodegree}[\cdot]$  values in linear time, given a graph in adjacency list format.

```
1  #!/usr/bin/python
2  def twodegree(arg):
3      for count, data in enumerate(arg):
4          counter = 0
5          for count2, data2 in enumerate(data):
6              counter += len(arg[data2])
7          print("twodegree[u] where u is", count, "is", counter, "degrees
8              .")
9
10 representation = [[1, 2], [0, 2], [0, 1, 3], [2]]
11 twodegree(representation)
```

The reason why the above algorithm works is it takes in the list already in adjacency list form, then for any given vertex  $u$ , it finds all the adjacent vertices, and then measures the length of their adjacent vertices list. Hence, it will take the amount of time to count all the edges (or every vertex connection a vertex has, which is equal to the total number of edges), + some constant integer for calculating the length of a list.