

# Problem set 1: Problem 5

- Due Wednesday by 11:59pm
- Points 0
- Submitting on paper
- Available Feb 3 at 12am - Feb 28 at 11:59pm

In this question, you are going to implement the pagerank algorithm.

Download the following mini [Stanford-web-graph](http://snap.stanford.edu/data/web-Stanford.html)  (<http://snap.stanford.edu/data/web-Stanford.html>) data set.

The data set is a text file in which each row is in the format

from\_id to\_id

which means there is a directed edge from the vertex from\_id to the vertex to\_id.

In class, we will go over how to approximate pageranks efficiently. The idea is as follows.

We start with  $\pi^{(0)} = (1/n, 1/n, \dots, 1/n)$ .


At iteration  $i$ , we update

$$\pi_v^{(t)} = (1 - \epsilon) \sum_{u \rightarrow v \in E} \frac{\pi_u^{(t-1)}}{d_u}.$$

If you run for  $T$  iterations where  $T = \log_{1-\epsilon} \delta$ , then the average error  $|\pi_v^{(T)} - \pi_v| \leq \delta$ .

Your goal is to implement the above iterative algorithm to estimate the pageranks (as opposed to using Gaussian elimination since this mini-graph is already very large). Your output would be a text file in which each line contains a nodes and it pagerank in decreasing order.

In the code, you set  $\delta = 0.00001$ .

You can use the following [C++ template](https://sdsu.instructure.com/courses/151706/files/12393296?wrap=1) (<https://sdsu.instructure.com/courses/151706/files/12393296?wrap=1>)  ([https://sdsu.instructure.com/courses/151706/files/12393296/download?download\\_frd=1](https://sdsu.instructure.com/courses/151706/files/12393296/download?download_frd=1)) (that I wrote) or implement it using any programming language of your choice (which might be significantly slower).

