Neo4J Commands

Below are the comands used in Neo4J.

Load data: Using import tool.
Credit (https://neo4j.com/docs/operations-manual/current/tutorial/import-tool/).
The Papers and Authors are nodes and their relationship are defined here.

```
bin/neo4j-admin import --nodes:Author=import/author.csv --
nodes:Paper=import/papercleaned.csv --
relationships:CITED=import/citation.csv --
relationships:AUTHOR=import/authorship.csv --ignore-missing-nodes
```

Graph schema:
How the graph looks, it is defined based on the above command.

```
call db.schema()
```

Search for paper by a title:
Create Index for faster search using paper title. This will improve search times.

```
CREATE INDEX ON :Paper(title);
```

Search for paper by a title:

```
MATCH (p:Paper)
WHERE p.title="Silent Data Corruption - Myth or reality?"
RETURN p
```

Search for papers containing keywords.
Credit (https://stackoverflow.com/questions/45212213/is-there-any-way-in-neo4j-using-contains-to-compare-case-insensitive-string)

```
MATCH (p:Paper)
WHERE toLower(p.title) CONTAINS toLower("Artificial Intelligence")
RETURN p
```

PageRank algorithm:
Credit (https://tbgraph.wordpress.com/2018/09/09/article-recommendation-system-on-a-citation-network-using-personalized-pagerank-and-neo4j/)
Credit (https://neo4j.com/docs/graph-algorithms/current/algorithms/page-rank/)

Calles the built in PageRank algorithm on the Load Data above. As we can see the nodes Paper and relationship CITED have been defined.

```
CALL algo.pageRank('Paper', 'CITED',
  {iterations:20, dampingFactor:0.85, write: true, writeProperty:"page_rank"})
YIELD nodes, iterations, loadMillis, computeMillis, writeMillis,
dampingFactor, write, writeProperty
```

Iterations: Number of times its run, default is 20.
dampningFactor: Described in more detail above in Section 2.x.x. Default 0.85.
write: If we should write the score to the node.
writePropoerty: What heading we will write it back too
loadMillis: Miliseconds to load the data
coputeMillis: Miliseconds to run the algorithm
writeMillis: Miliseconds to write the result.
ArticleRank algorithm:
Credit (https://neo4j.com/docs/graph-algorithms/current/algorithms/article-rank/)
Very similar to PageRank above except built in algorithm is different.

```
CALL algo.articleRank('Paper', 'CITED',
  {iterations:20, dampingFactor:0.85, write: true, writeProperty:"article_rank"})
YIELD nodes, iterations, loadMillis, computeMillis, writeMillis,
dampingFactor, write, writeProperty
```

Top 10 PageRank Results:
Credit (https://tbgraph.wordpress.com/2018/09/09/article-recommendation-system-on-a-citation-network-using-personalized-pagerank-and-neo4j/)
As we can see from above in the PageRank command the write_property is page_rank. This is the score written to the node. So here we are just calling this property, assigning it a title and displaying the top 10 in order of descent.
Here result displayed as a table using the return with Title, Year and CitationCount in the Table.

```
MATCH (p:Paper)
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations, p.
num_papers_cited as NumPapersCited, p.page_rank as PageRank
ORDER BY PageRank DESC
LIMIT 10
```

Top 10 ArticleRank Results:

```
MATCH (p:Paper)
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations, p.
num_papers_cited as NumPapersCited, p.article_rank as ArticleRank
ORDER BY ArticleRank DESC
LIMIT 10
```

Top 10 PageRank papers based on a key word.
Building ontop of above commands:

```
MATCH (p:Paper)
WHERE toLower(p.title) CONTAINS toLower("Artificial Intelligence")
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations,
p.page_rank as PageRank
ORDER BY PageRank DESC
LIMIT 10
```

Top 10 ArticleRank based on keyword:

```
MATCH (p:Paper)
WHERE toLower(p.title) CONTAINS toLower("Artificial Intelligence")
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations,
p.article_rank as ArticleRank
ORDER BY ArticleRank DESC
LIMIT 10
```

Individual Paper citation count

```
MATCH (p:Paper)
WHERE p.title="Silent Data Corruption - Myth or reality?"
RETURN p.title AS Title, p.num_citations as NumCitations
ORDER BY NumCitations DESC
LIMIT 10
```

Top 10 Citation Counts:

```
MATCH (p:Paper)
RETURN p.title AS Title, p.year As Year, p.num_citations as NumCitations, p.
num_papers_cited as NumPapersCited
ORDER BY NumCitations DESC
LIMIT 10
```

1500-2500 PageRank papers:

```
MATCH (p:Paper)
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations, p.
num_papers_cited as NumPapersCited, p.page_rank as PageRank
ORDER BY PageRank DESC
SKIP 1500
LIMIT 1000
```

2250-2750 ArticleRank papers:

```
MATCH (p:Paper)
RETURN p.title AS Title, p.year as Year, p.num_citations as NumCitations, p.
num_papers_cited as NumPapersCited, p.article_rank as ArticleRank
ORDER BY ArticleRank DESC
SKIP 2250
LIMIT 500
```