

Counting shared nearest neighbors

IN4200 - Home exam 1, Spring 2021

Information on the files, and how to compile them

Here is a brief overview of the files

- `read_graph_from_file1.c` Reads the graph from file and stores in 2D table(matrix) format
- `read_graph_from_file2.c` Reads the graph from file and stores in CRS format, contains functions `merge` and `mergesort` as well, these are used in sorting the `col_idx` array, the functions are used inside the function `read_graph_from_file2`.
- `create_SNN_graph1.c` Creates a shared nearest neighbors(SNN) graph, in 2D table format
- `create_SNN_graph2.c` Creates a shared nearest neighbors(SNN) graph, in CRS format
- `check_node.c` Checks whether a given node can be in a cluster, also contains the function `dfs`, this function is used inside the function `check_node`
- `functions.h` Header file, contains all the functions. This file also contains some printing functions that I used while I was developing the code, however these functions are not called when testing for larger files
- `graph.txt` Simple graph file(from example graph in exam text), used to check my functions while I was developing my code
- `facebook.txt` Large graph file(given in exam text), I will be using this to run and find the timings of the functions
- `main_2Dtable.c` Main file, contains all the test functions for the graph in 2D table format
- `main_CRS.c` Main file, contains all the test functions for the graph in CRS format
- `makefile` Makefile to make it easier to run the codes(see decription below for details)

Code compilation

Here are the instructions on how to compile the codes. I have made two main files, one main file tests all the relevant functions in the 2D table case, and one test all the relevant functions for the compressed row storage(CRS) case. I have added a makefile. **NOTE** I have run the codes with the input file `facebook.txt`, If you wish to change the input file then go into the makefile, and type the name of the file in the line `./a.out facebook.txt`, i.e, it would be like `./a.out <filename.txt>`.

To test the functions for the 2D table(no parallelization)

```
make main_2D
```

To test the functions for the 2D table, with openMP **parallelization**

```
make main_2Domp
```

To test the functions for the CRS format(no parallelization)

```
make main_CRS
```

To test the functions for the CRS format, with openMP **parallelization**

```
make main_CRScomp
```

Test Output

Here I have added an example of the output I receive when I run all the main files, I have run all the main files with the input file `facebook.txt`. All the output that I have written here, were produced by running my code on ifi's Linux servers, that is, I used ssh login.ifi.uio.no, this was also mentioned in the home exam text.

Output I get when I run `make main_2D`:

```
[terminal]$ make main_2D
gcc main_2Dtable.c read_graph_from_file1.c read_graph_from_file2.c create_SNN_graph1.c create_SNN_graph2.c
./a.out facebook.txt
```

Timings for connectivity graph in 2D table format
(without parallelization)

```
Connectivity graph information
-----
Total no. of nodes N, are: 4039
Total no. of edges E, are: 176468
```

```
(Edges read from file are 88234)
-----
```

```
Time taken for read_graph_from_file1 = 0.089087 seconds
-----
```

```
SNN 2D table allocated of size: 4039 x 4039
-----
```

```
Time taken create_SNN_graph1 = 1.325817 seconds
-----
```

Output I get when I run `make main_2Domp` :

```
[terminal]$ make main_2Domp
gcc -fopenmp main_2Dtable.c read_graph_from_file1.c read_graph_from_file2.c create_SNN_graph1.c create_SNN_graph2.c
./a.out facebook.txt
```

```
Timings for connectivity graph in 2D table format
(with parallelization)
```

```
Connectivity graph information
-----
```

```
Total no. of nodes N, are: 4039
Total no. of edges E, are: 176468
(Edges read from file are 88234)
-----
```

```
Time taken for read_graph_from_file1 = 0.092542 seconds
-----
```

```
SNN 2D table allocated of size: 4039 x 4039
-----
```

```
Time taken create_SNN_graph1 = 0.348335 seconds
-----
```

Output I get when I run `make main_CRS` :

```
[terminal]$ make main_CRS
gcc main_CRS.c read_graph_from_file1.c read_graph_from_file2.c create_SNN_graph1.c create_SNN_graph2.c check_node.c
./a.out facebook.txt
```

```
Timings for connectivity graph in CRS format
(without parallelization)
```

```
Connectivity graph information
-----
```

```
Total no. of nodes N, are: 4039
Total no. of edges E, are: 176468
(Edges read from are file 88234)
-----
```

```
Time taken for read_graph_from_file2 = 2.066268 seconds
-----
```

```
The length of col_idx is: 176468
The length of row_ptr is: 4039
The length of SNN_val is: 176468
-----
```

```
Time taken for create_SNN_graph2 = 6.164284 seconds
-----
```

```
Here are the other nodes in a cluster with node:600, and tau:100
{}
-----
```

```
Time taken for check_node = 0.000032 seconds
-----
```

Output I get when I run `make main_CRSomp` :

```
[terminal]$ make main_CRSomp
gcc -fopenmp main_CRS.c read_graph_from_file1.c read_graph_from_file2.c create_SNN_graph1.c create_SNN_graph2.c check_node.c
./a.out facebook.txt
```

```
Timings for connectivity graph in CRS format
(with parallelization)
```

```
Connectivity graph information
-----
```

```
Total no. of nodes N, are: 4039
```

```
Total no. of edges E, are: 176468
(Edges read from are file 88234)
-----
```

```
Time taken for read_graph_from_file2 = 2.093909 seconds
-----
The length of col_idx is: 176468
The length of row_ptr is: 4039
The length of SNN_val is: 176468
-----
Time taken for create_SNN_graph2 = 0.872270 seconds
-----
```

```
Here are the other nodes in a cluster with node:600, and tau:100
{}
```

```
-----
Time taken for check_node = 0.000118 seconds
-----
```

Description of the check_node function

As you can see in the example run above, I run the check_node function with node_id=600, and tau=100, this gives an empty list as there are no clusters formed for these values. Since the facebook.txt is quite large, it is hard to test check_node on it. To demonstrate my check_node function, I ran the function with the graph.txt file, with node=1, and tau=2, the result I got was { 0, 2, 3, }, which is what I expected. Here is the the output I get,

```
[terminal]$ make main_CRSSomp
gcc -fopenmp main_CRS.c read_graph_from_file1.c read_graph_from_file2.c create_SNN_graph1.c create_SNN_graph2.c check_node.c
./a.out graph.txt
```

```
Timings for connectivity graph in CRS format
(with parallelization)
```

```
Connectivity graph information
-----
Total no. of nodes N, are: 5
Total no. of edges E, are: 16
(Edges read from are file 8)
-----
```

```
Time taken for read_graph_from_file2 = 0.000358 seconds
-----
The length of col_idx is: 16
The length of row_ptr is: 5
The length of SNN_val is: 16
-----
Time taken for create_SNN_graph2 = 0.008750 seconds
-----
```

```
Here are the other nodes in a cluster with node:1, and tau:2
{ 0, 2, 3, }
```

```
-----
Time taken for check_node = 0.000017 seconds
-----
```

If you wish to change the input node_id and tau value for the check_node function, then you can do so by going into the main_CRS.c file, and entering your choice manually in the check_node function. The first argument is the node_id, and the second argument is the tau value, all the other arguments for the check_node function remain unchanged.