Part III - BFS implementation

Now, we require you to implement a parallel BFS algorithm for undirected graphs, which calculates the shortest path matrix for the given graph (note: to represent infinity, mark the distance as -1).

The graph will be inputted exactly like the DFS algorithm from practical session number 5. We recommend you to test your code using input redirection from a file, so you won't have to input a graph each time.

The signature for the function should be as follows:

```
void bfs(Graph *graph, int **m);
```

Where graph is the given graph, and m is the shortest path matrix you need to update its values.

For this, you can write your code in either c or cpp. And you will need to submit a makefile. We will provide 2 makefiles that you could use (although you could also provide your own makefile), one for c, and the other for cpp.

Please note that you must use either <pthread.h> or <thread> as your thread library. Please also use a thread pool, you are permitted to use the code from the 5th practical session.

Note: you may change the SyncGraph implementation to suit your needs better, as long as the program still compiles and runs fine. However, you cannot change the implementation of main.c