

---

Parallel Computing  
Assignment 1 (15 Marks)  
Deadline: 18th October 11:59PM

---

- Implement shortest path algorithm using matrix multiplication. Assume an undirected graph with edge weights 1. Please refer to `shortest_path.py` for the algorithm.
  - You can generate a random graph and dump the same as an adjacency matrix to a file. The file can be assumed to be accessible to all processes.
  - The number of processes is a perfect square. Your code will be tested on (1, 2, 4, 16, ...) processes. The number of nodes in the graph can be assumed to be a multiple of number of processes.
  - Output is the shortest path.
  - Add code to verify your output.
  - Time the runtime of your algorithm (excluding the file write/read) for various runtime parameters.
  - Use github to host your code. Share the link as your submission. Any other files/zip-files attached will not be evaluated.
  - Any kind of cheating will be heavily penalized. Please refer to the discussion we had in class regarding this.
  - The best assignment submission will fetch you a prize.
  - If you need any help or have any questions, feel free to shoot me an email. By any, I mean, any. For example: Why did Kattappa kill Baahubali? I will try to answer, some answer, though might not be the right answer!
  - If you need help in MPI installation, our hero Ali has already got it working on his laptop, and might be able to help you for some commission/fees.
  - Two good resources: [mpitutorial.com](http://mpitutorial.com) and [google.com](http://google.com) (though I would suggest you to use [duckduckgo.com](http://duckduckgo.com))
  - Clash on!
1. The adjacency matrix can be assumed to fit on a single process (6 marks).
  2. The adjacency matrix cannot be allocated on a single process, but will occupy all the processes on which the program is launched on (15 marks). Please note that the maximum marks you can get in this assignment is 15. You can choose to skip the first question, and only attempt this.