Frequently Asked Questions (FAQs)

1. Due Date / Time?
   1. Not later than 23:59 hours on Friday 22nd of November, 2019.
2. Where to submit the assignment?
   1. Google form: <https://forms.gle/NbxK12dNNVQfiMLp8>
   2. Both the c file and txt file (report) should be submitted in one shot.
   3. Only one submission, no re-submits.
3. How is the evaluation done?
   1. By running a Python script on the set of submission files. That is why file name with your SRN is extremely important.
4. How to compile and test after I implement the functions in PES1201800000.c?
   1. gcc -Wall graphprofile\_test.c PES1201800000.c
   2. ./a.out < sample\_test.txt
   3. Make sure all tests passed.
   4. This is only a sample test. The actual evaluation is going to have different graphs as input, which is hidden from you.
   5. Any compilation error or runtime error leads to zero marks.
5. Can I modify the header file or the graphprofile\_test.c?
   1. No, you need to submit only PES1201800000.c (implementation file) and PES1201800000.txt (report file). We are going to run against different test file with other graph examples.
6. How are the marks allocated?
   1. 8 marks for the PES1201800000.c (implementation file) and 2 marks for the PES1201800000.txt (report file).
7. Can we define our own functions other than the 8 functions to be implemented?
   1. Yes, you can have your own private functions too. Just make them “static”. Prepend your function definition with the keyword “static” so that it’s not visible outside of your c file and won’t conflict with a function with possibly the same name in our test file.
8. Can we change the signature of some of the functions to be implemented?
   1. No, our tests are depending on the signature of the 8 functions. But, you can have your own private (static) functions which can be called from the public (our 8) functions.
9. Can we have global variables defined in the implementation file?
   1. Avoid global variables at all times. Instead, pass a reference (pointer) to a variable which acts as a global variable.
10. What does constraints mean?
    1. Constraints are ensured by us, so you don’t have to worry about the cases outside of the constraints. E.g. if the constraint says the graph is connected, you can safely assume all our testcases has connected graphs.
11. Constraints say n is in the range [3, 100]. What does it mean?
    1. The testcases we are going to test your code on will not have n less than 3 or greater than 100, so you don’t have to check for the cases with n less than 3 and your code doesn’t have to efficient to run for more than n=100.
12. What if my implementation is inefficient?
    1. We have a timelimit of 1 second for your implementation. If the innermost operation of your functions does not execute more than n2 number of times, you don’t have to worry about the efficiency. In other words, make sure your functions execute in O(n2) time.
13. Do we have to compile with -Wall option?
    1. Doing so makes sure you don’t leave any warnings unattended. Typically, warnings are an indication of semantic/run-time errors which pop-up only for certain testcases. Fixing the warnings reduces the runtime errors and chance of you getting zero marks reduces.
14. Making sure our code works for the given sample tests is good enough to ensure 100% marks?
    1. No, sample tests help you to make sure your compiles properly and runs for typical cases. The testcases we run are more exhaustive and your implementation may fail in some cases. We won’t share the testcases we use for the assessment.
15. “Is the graph a cycle graph?” If the graph has a cycle, is it called a cycle graph?
    1. The question is about “the graph **is** a cycle graph”.