

Roll number	Technical correctness (40)	Object oriented design (10)	Total (50)	Comments
cs1401	0	0	0	No Submission
cs1402	0	0	0	No Submission
cs1403	0	0	0	No Submission
cs1404	5	2	7	The code does not compile and work. Readme does not explain anything also. Some effort in calculating number of neighbors is evident, thus some marks.
cs1405	22	2	24	Your code is computing the neighborhood profile only for two levels. Also, the code is not commented at all. Please use parametrized types Iterator<T>, ArrayList<T> instead of raw types.
cs1406	0	0	0	No Submission
cs1407	0	0	0	No Submission
cs1408	0	0	0	Almost blank code
cs1409	10	8	18	You seem to have got the algorithm right, but the code has compilation issues at several points which we could not fix even after trying to change some part of your code.
cs1410	13	8	21	You seem to have got the algorithm right, but the code fails with queue underflow.
cs1411	35	8	43	Good working simple code. However, the method specified should have returned the arraylist of numbers, not members! No need to give nice print statements, return the required object properly instead.
cs1412	0	0	0	No Submission
cs1413	10	6	16	The algorithm has something wrong. The neighborhood profile should be monotonically increasing, but your code gives wrong results. You seem to have understood the idea though.
cs1414	10	0	10	You have written so much of code to directly read data from the files, but have not checked that the provided classes already do it and you were supposed to just use them!! The idea of using BFS is right, but the code produces an array with all zeros.
cs1415	0	0	0	No Submission
cs1416	10	8	18	The idea is right, but the program probably goes into an infinite loop, as it does not finish.
cs1417	38	8	46	Good code. Minor mistake: you missed out in calculating the member itself.
cs1418	12	8	20	Good design and right idea, but the results are very wrong.
cs1419	0	0	0	No Submission
cs1420	0	0	0	Nothing in the code.
cs1421	5	2	7	The code does not run, has compilation problems, is incomplete too. However, the basic idea of the algorithm is written in some comments.
cs1422	0	0	0	No Submission
cs1423	0	0	0	Almost blank code.
cs1424	5	4	9	The program crashes with NullPointerException. Not clear if you understood the idea of BFS right as well.

cs1425	0	0	0	No Submission
cs1426	38	9	47	Good design, good code. Just that you are counting $ N(a,1) $ twice.
cs1427	0	0	0	No Submission
cs1428	0	0	0	No Submission
cs1429	3	1	4	Wrong algorithm, wrong results.
cs1430	10	6	16	The code crashes with concurrent modification exception for all the inputs I tried. You seem to have got the idea right and the design is fine.
				General remark: since BFS is a general useful module, why not write it in a separate class, or at least as a public method in the Utils class?