

Project Assignment

This is an individual/group project. Groups of **up to three** students can submit joint solutions. For group submissions, please follow the following rules:

Notes regarding groups (If you are submitting the work individually, please skip this part):

For group submissions, please follow the following rules:

- 1) **Form a Group AGAIN:** First, self-enroll in a group with your partner(s) by navigating to People → Groups → **Assignment4 Group**.
- 2) **Re-enrollment Required:** Even if you were part of a group before, you must sign up for the Assignment3 Group again, as it is a separate tab under the Groups section.
- 3) **Submit as a Group:** When submitting the assignment, select the "Group Submission" option. This allows one team member to submit the work on behalf of the entire group, ensuring everyone in the group sees the grade and feedback. Important: **ALL GROUP PARTNER NAMES** should appear on ALL submitted documents.

You may use the Canvas Discussion Board to look for groupmates and create your group.

As it appears in the course syllabus, for the homework assignments, students are encouraged to discuss the problems with others, but you are expected to turn in the results of your own effort (not the results of a friend's efforts)". Even when not explicitly asked, you are supposed to justify your answers concisely.

Question 1)

Given the root of a **binary tree**, write a **recursive** program that counts the number of nodes at even levels of the binary tree. A node at level 0 is considered to be at an even level. The root node is at level 0, its children are at level 1, its grandchildren are at level 2, and so on.

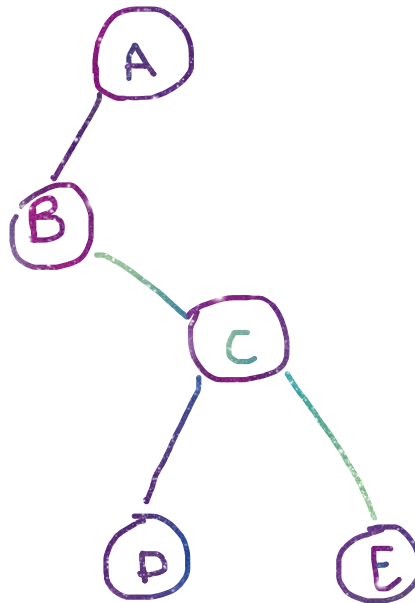
Tasks: (1) Implement a **complete program** in either C++ or Java that **recursively** counts the number of nodes located only at **even** levels of a binary tree. Your program should be able to read an arbitrary binary tree as input from the user. No points for any other programming languages except for C++/Java.

(2) What is the time complexity of your algorithm in the worst case when the binary tree has n nodes? Analyze and discuss the reasoning clearly. First write the corresponding **recurrence relation** then solve it to express it in Big-O notation. Show all your work.

Requirements:

1. The program should handle arbitrary binary trees where not every parent has two children (some nodes may only have one child). Ensure your code covers all possible cases.
2. The program should read the binary tree as input in a user-defined format. **Do not ask for clarification on how to format the input** please. Make sure your approach is outlined in a separate **README** file, detailing how the binary tree is read and processed.

Example of Binary Tree:



Expected Output: For this example, the nodes at even levels are A and C. The count of nodes at even levels is 2.

README Instructions:

- Explain how the binary tree input should be provided and processed.

Deliverables:

- 1) You need to submit **all your program files** on Canvas in the related digital repository (located under the Assignments section) by the specified deadline. The program should be executed without any issue otherwise no points will be assigned to this submission. **Do NOT zip the files.**
- 2) As a project **report**, in addition to the code files, **paste your finalized code in the report file** (50% of the obtained grade will be dropped if this part is missed in the submitted report file). The report also should contain the **analysis** of the worst- case time complexity for

your programs as requested. Students are expected to show all their work and do not skip the details.

3) README file for input format details.

Important NOTE: Don't share your solutions with others to avoid academic dishonesty. All submissions will be checked for that. Academic dishonesty and/or plagiarism in this course will result in a failure of the assignment, or the course, plus an AD report to the department and the university. Please be careful and present the result of your own effort.