

EEL4837 Programming for Electrical Engineers II – Spring 2023**Excursion 2: Technology Mapping****Due Date:** 11:59 PM, Sunday, April 26, 2023**Submission Instructions**

Submit your code on Gradescope. You need to submit the following files **separately (not as an archive)**:

1. One or several source code files.
 - Your program should be able to read an input file named ***input.txt*** located in the same directory as your source code.
 - Your output should be a single integer, which should be written to a file named ***output.txt*** and located in the same directory.
2. A documentation named ***readme.txt***.
 - Explicitly indicate if you are claiming bonus points.
 - (Optional) Any other descriptions of your program if you wish to provide.

Description

The goal is to implement an elementary tool for technology mapping, which will:

1. Transform the original logical expressions into a NAND-NOT tree;
2. Calculate the minimal cost of the transformed structure.

Input: a Boolean model specified in the format as shown in the Excursion 2 slides.

Output: a single integer, which is the calculated minimal cost of the transformed structure.

You may assume:

1. Each node of the input netlist is a *single* AND, OR, and NOT function.
2. In each test case, there will be one output node only. That is, your circuit will be a tree.
3. A Technology Library is available, which consists of the components listed in Table 1.

Table 1. Technology Library

Name	Cost
NOT	2
NAND2	3
AND2	4
NOR2	6
OR2	4
AOI21	7
AOI22	7

Guidelines

1. Read a netlist as input.
2. Parse the netlist to construct a tree based on the provided expressions.
3. Convert the tree into a NAND-NOT tree.
4. With the reference to the Technology Library (Table 1), recursively compute the minimal cost at each node.
5. Write the final minimal cost to the output file.

Grading

Total points: 100

1. Automated grading (75 pts)
 - a. Correctness (60 pts): correct output values on well-formed, “legal” inputs.
 - b. Robustness (15 pts): correct output on very simple and very complex “legal” inputs.
2. Manual grading (25 pts)
 - a. Efficiency: the code is reasonable and not brute-force, no particular time/space complexity is expected though.

- b. Elegance: clean code style, readable comments for key steps as needed, meaningful functions, etc.
3. Bonus points (20 pts): implement a dynamic programming approach that traverses the tree bottom-up and saves the intermediate costs for subtrees.

Late Policies

- Late due date: 10 days after the due date.
- 5% of the max total points per late day will be deducted.

Notes

1. Your code should be in C++ only. We encourage splitting up your code into multiple files/functions/classes.
2. Your program should compile and run successfully on both the ECE Linux server and Gradescope.
3. You can only use C++ standard libraries (including STL). Do **not** use libraries that implement circuits, trees, or graphs for you. If you are not sure whether or not some libraries are allowed, please reach out to the TA or instructor for help.