ITU Computer Engineering Department

BLG 223E Data Structures, Summer 2021

Homework #5

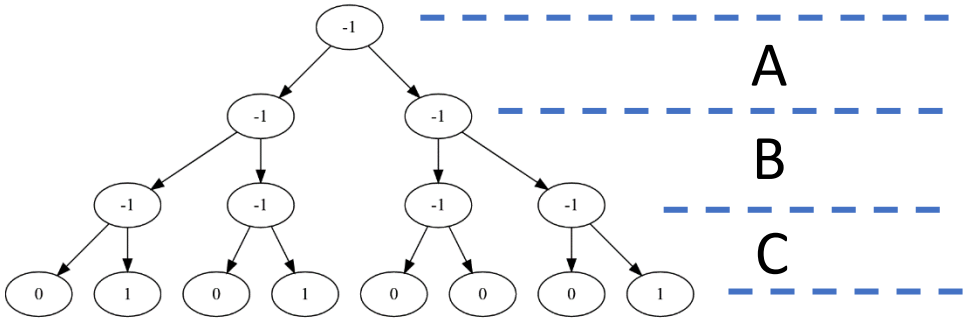
Due August 3, 2021 23:59

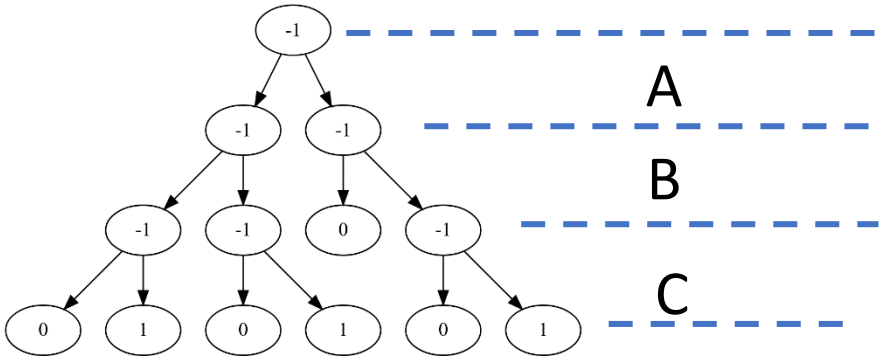
Definition

You are asked to implement a program that finds the variable ordering for a boolean function that can be represented with a reduced binary decision tree that contains minimum number of nodes.

Below given a boolean function with its full and reduced binary decision tree, and another reduced binary decision tre efor a different variable ordering.

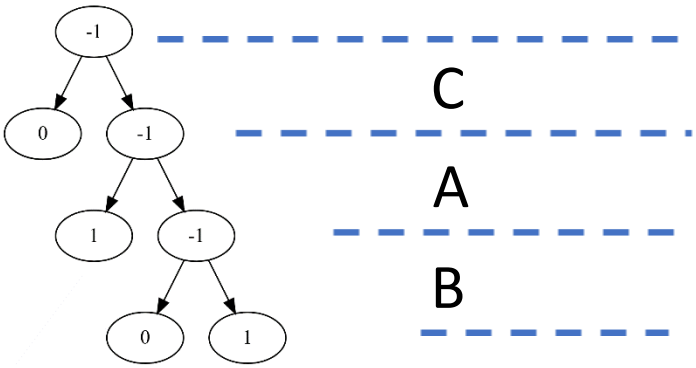
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1





The same function for a different variable ordering:

C	A	B	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1



In the diagram above, inner nodes have been shown to keep -1 as data just for informative purposes. You may choose to mark an inner node using an alternative approach.

Please pay attention to the following details in your implementations:

- In your program, the initial inout would always belong to the ordering of dictionary ordered capital single letters form the English alphabet e.g. “ABCDEF…”
- A hint for your program run can be given as follows:
    1. Produce a truth table for the provided input
    2. Produce the outputs for different variable orderings. Pay attention that different orderings can be found with the help of a recursive permutation function. Another important point is that you need to use truth table in step 1 to calculate the outputs but you do need to save only the outputs not the whole truth table.
    3. For each combination in step 2 build a binary decision tree using the output. This can be done using a preorder construction.
    4. For each tree built in step 2, reduce the tree by collapsing the subtrees that provide a single output (see Example above). This can be dome using a number of postorder traversals.
    5. Count the number of nodes in the tree (Using any kind of traversal)
- **You are asked to implement a binary tree data structure, other possible solutions will not be graded!!!**
- Using global variables is alright for this homework.
- A large timeout value will be given, so please do not focus on the efficiency of your code.
- In test cases, output for a maximum of an 8 variable function will be given (2^8 binary digits). You may test your code for 9 variable functions and analyze the performance as an exercise.

Input-Output

Your program is going to accept a single binary string from the command line without spaces and produce and output the ordering(s) that produces the minimal tree and the number of nodes in that tree. If there are multiple orderings, they should be printed in alphabetical order. The input will be syntactically and semantically correct and the number of digits it contain will always be a power of two.

For instance, for the above example it should accept the following input form the command line:

01010001

And it will produce the following out to the terminal:

CAB CBA 7

Deliver

Please zip and deliver the directory structure defined below:

- HW5: Topmost folder, that will contain all the folders in your submission. No other files should be present under this folder in your submission.
- HW5/src: Contains all the *.cpp files
- HW5/src/main.cpp: Contains your main function and other code you want to deliver.
- HW5/src/Node.cpp: Contains the code for your binary tree node.
- HW5/src/Tree.cpp: Contains the code for your binary tree.
- HW5/include: Contains all the header files you use
- HW5/bin: An empty directory that will contain objective files when your project is compiled

Please check the calico test file in the homework definition to see how your files will be compiled and tested.

Restrictions and Guidelines

- Compilation environment: Only the code that can be compiled by the environment of the container definition provided in ninova will be accepted.
- Testing of your program will be performed using Calico (https://calico.readthedocs.io/en/latest/). Test cases that will be used to test your homework is provided as an attachment in ninova.
- **STL usage is allowed except for the binary decision tree data structure. Especially Bitset library (#include <bitset>) might be handy.**
- **This homework is for individual submissions**. Any kind of code sharing or code adaptation from an external source is strictly forbidden. Submitted code will undergo a plagiarism check process and plagiarsim penalties may be given if the submitted code's similarity is approved by the instuctor.
- Make sure you write your name and number in all of the files of your project, in the following format:
  /* @Author
  Student Name:<studentname>
  Student ID :<studentid>
  Date:<date>*/
- Only electronic submissions through Ninova will be accepted no later than deadline
- Use comments wherever necessary in your code to explain what you did.