**Decision Tree Algorithm**

**Melvin Ismanto**

**9018720198**

**Algorithm Summary**

My program uses the concept of Decision Tree algorithm to extract/mine the frequent pattern from the database (text input files), so the big picture of the program should be easy to understand. The **main** steps of the algorithm is as follows:

1. Scan the dataset text file and move it to a local variable.
2. Create a Decision Tree (DT) node as a root.
3. Push the root node to a queue, *Q*.
4. While the queue *Q* is not empty, do step 5 until 12.
5. Scan the dataset parameter and count the occurrence of each attribute’s and classifier’s label.
6. Calculate Info(D) by using the formula -.
7. Find the maximum: , by finding the minimum: , and push each label of the selected attribute to a priority queue, *PQ*.
8. For each label on the selected attribute for partition, while *PQ* is not empty, create a DT child node.
9. Set the parent of the child node as current node.
10. For each child node, if its contribution to the Information Gain is more than 0, and there are still remaining attributes for further partitioning, filter the parameter dataset with the selected attribute equals to current attribute label, and create a new dataset based on that.

Then, push it to the queue, *Q*.

1. Find the majority classifier label for current attribute label
2. Pop the priority queue, *PQ*.

For more in-depth explanation of the algorithm and an easier understanding of each section of the algorithm, please refer to the comments provided on the DecisionTree.cpp file.

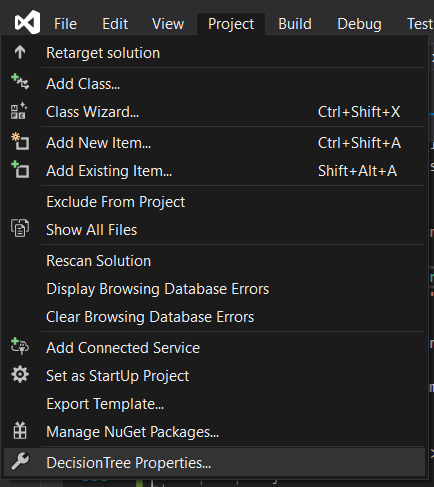
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

I uses the Microsoft Visual Studio Community **2019** (Version 16.0.1) IDE to write my code, using the C++ language. So you can also use that to make compiling easier.

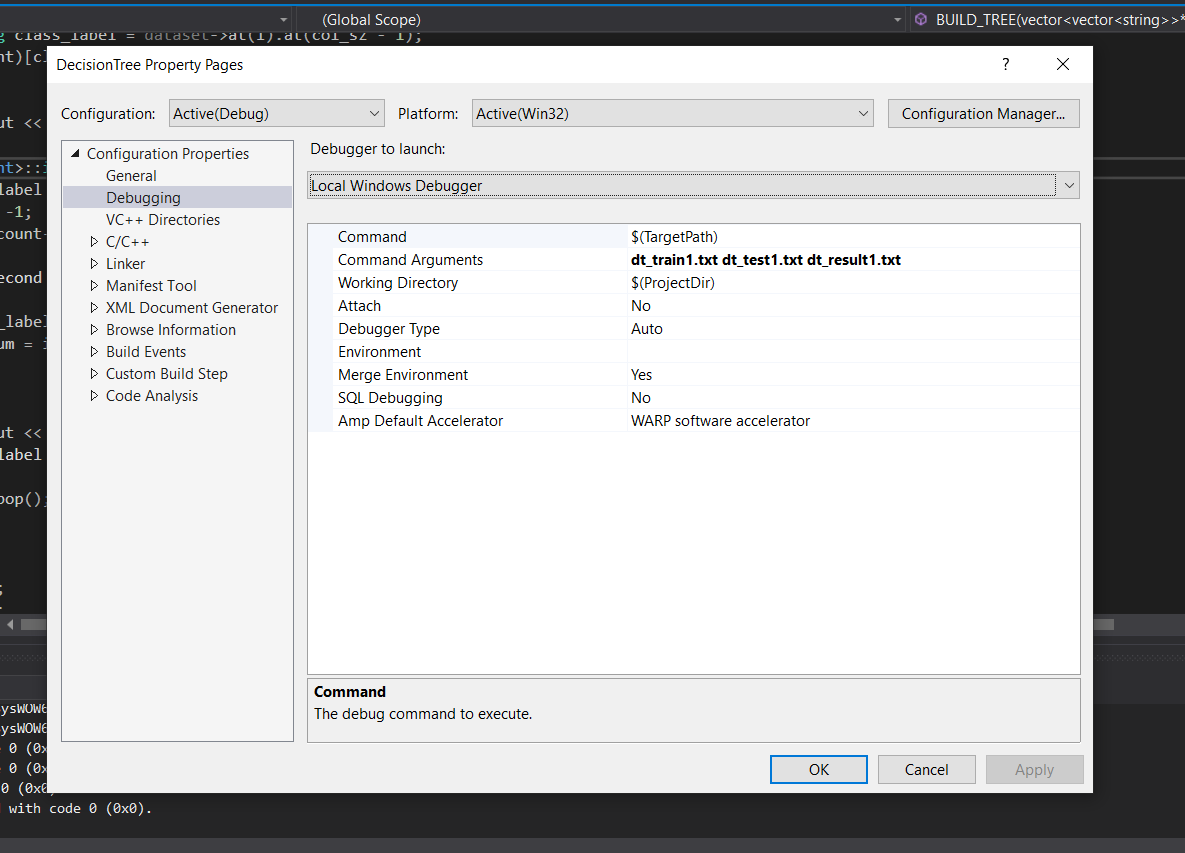
Download it here:

<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16>

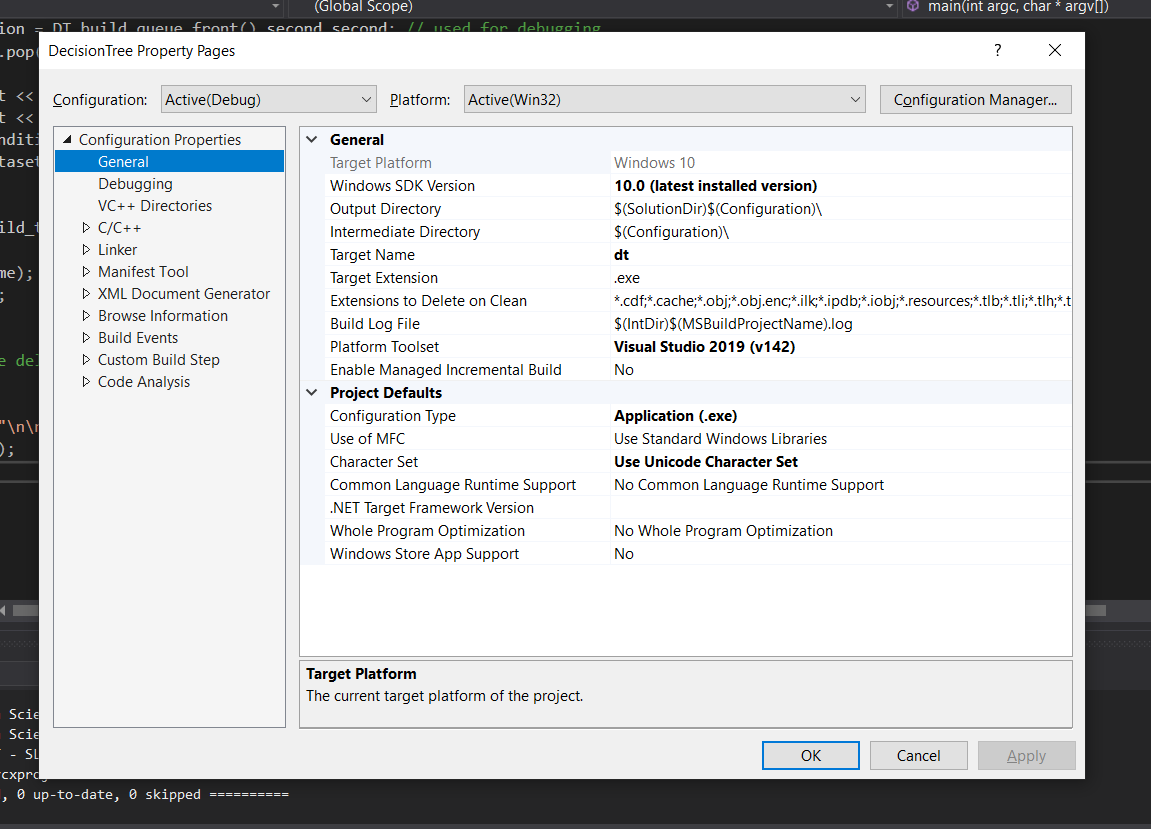
To run with parameter on VSC2019, go to Project > {project\_name} Properties…



Then,



To change the .exe output file, go to the Project properties, General, Target Name.



The .exe files with the VSC2019 (after debugging, NOT release) can be found on the “Debug” folder inside the project folder.

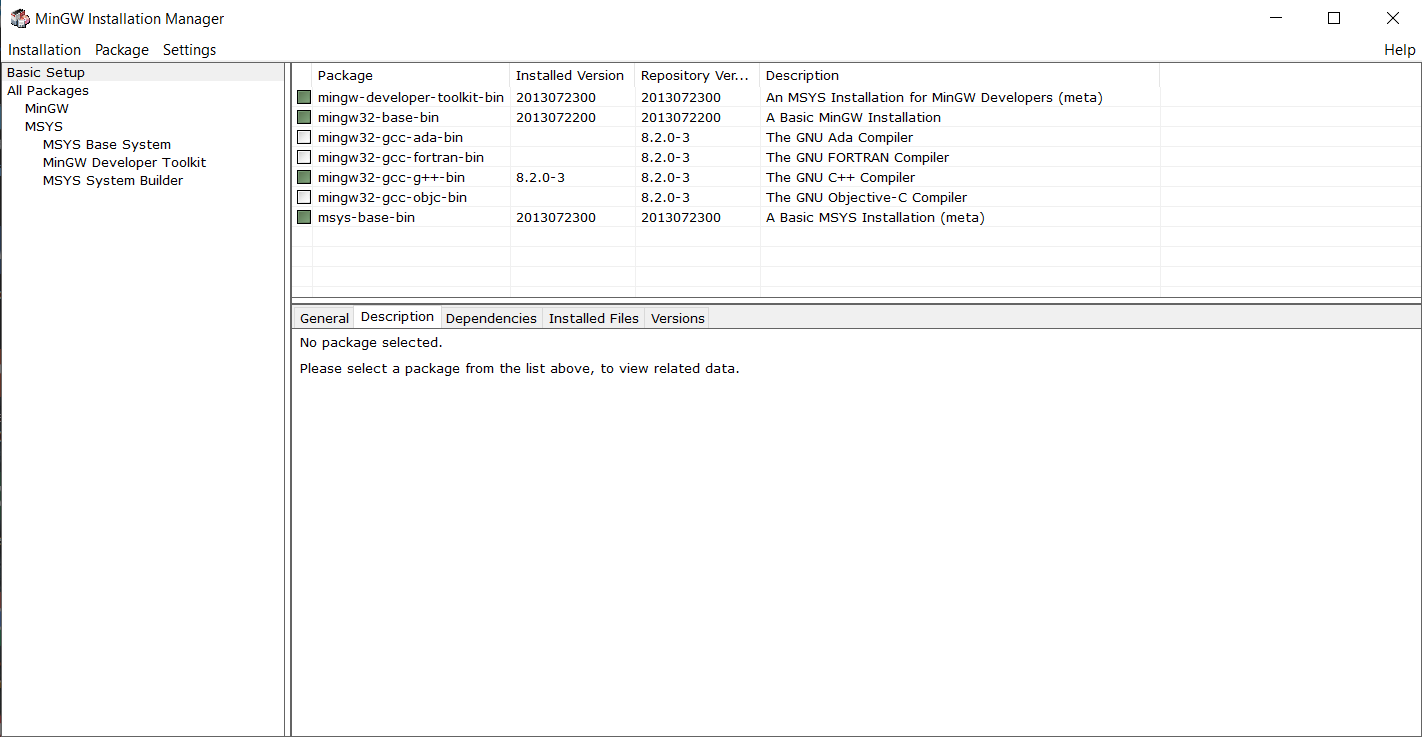
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

You can also try to use CodeBlocks (I uses version 17.12) with the MinGW (Minimalist GNU) GNU GCC, to compile the code. You can just take the Apriori.cpp source file from the project folder, and ignore the rest of the file.

You can get the MinGW installation here: <https://osdn.net/projects/mingw/downloads/68260/mingw-get-setup.exe/>

Then what you need to do is to open the file, uses the default option to install the MinGW (install to C:\MinGW), and open the shortcut on the desktop.

Choose THESE files:



And then Installation>Apply Changes

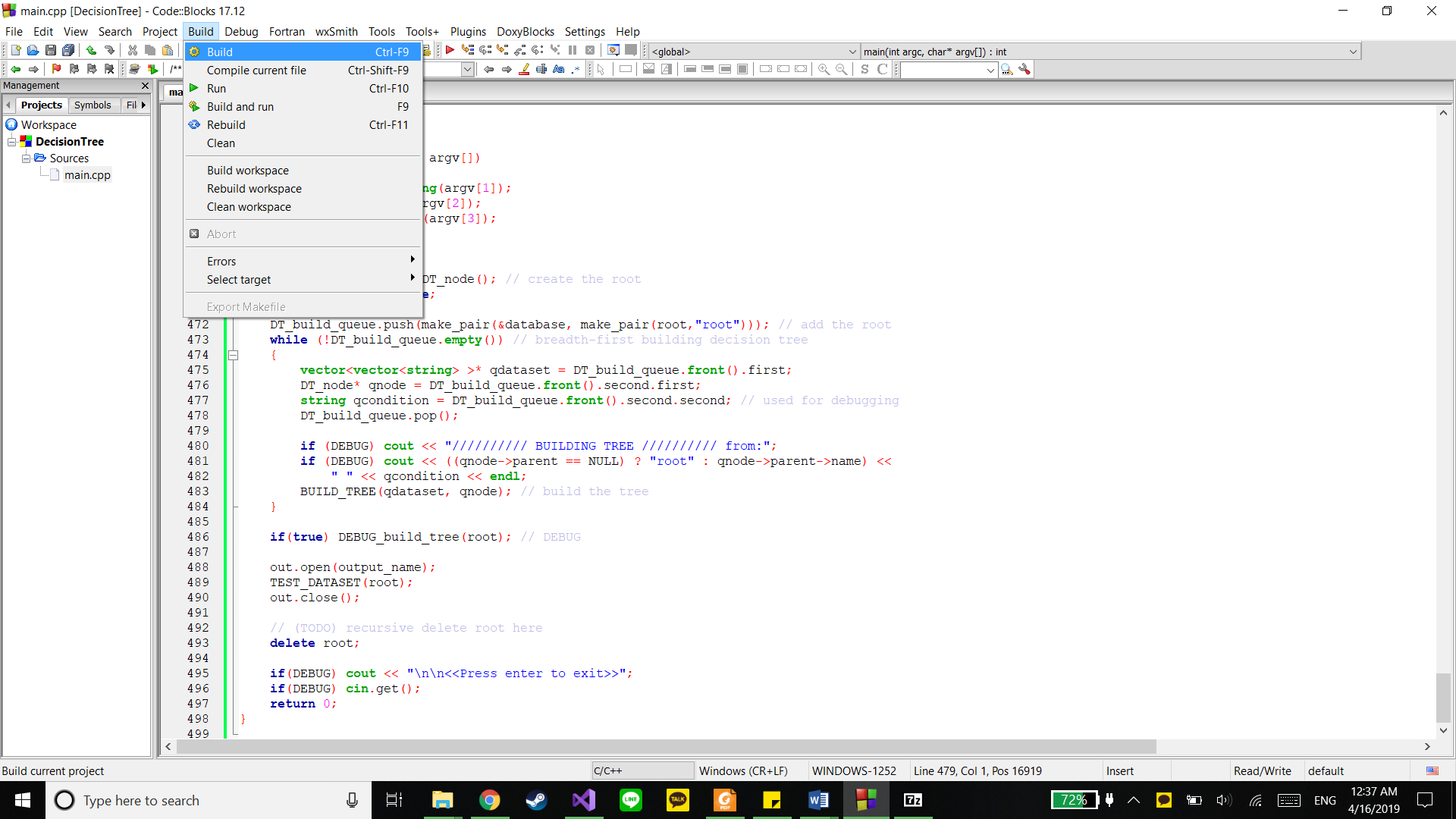
It will download the necessary files to compile C++ source files.

Then just download CodeBlocks from here:

<http://www.codeblocks.org/downloads/26>

And follow the installation procedure. You can now open the dt.cpp file with CodeBlocks

To build the source file, press ctrl+f9, or go to here:



.exe files after building CodeBlocks can be found on the same folder as the .cpp file.

After that, the .exe file can be run like how it’s mentioned on the assignment’s pdf file.

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

I code this on a Windows 10 Pro 1809 (64 bit), with Intel i7-7500 CPU, and 16GB of RAM on my Laptop. It takes about 1 second for my program to analyze the provided text input file, and output the result to a text file.