User Guide

CSE 360

Fall 2018

Ian Bradley
Melanie Hendricks
Cole Stanfield

Table of Contents

Introduction:

Thank you for choosing NAP Software. This document will cover the functionality of the program, providing installation instructions, a quick start guide, interface overview, and example runs. This useful guide provides a list of resources below to get started.

Program Overview:

The NAP program analyzes a network diagram and determines all paths in a network. It takes multiple occurrences of an activity name, duration, and list of predecessors (all entered manually or listed in a .csv file) and parses this data to create a network diagram. It then determines each path in the network, and lists out the name of each activity in that path, along with the total duration of that path, in descending order. The user can then download the list of paths as a .txt or .pdf file.

NAP also provides useful error checking. It can identify invalid inputs, unconnected nodes, and network loops. This information is displayed in the main window so the user can easily identify any mistakes.

Installation:

Before installation, make sure at least 10 MB of storage is available. The NAP installer can be started by double clicking its executable file. This will automatically install all components of NAP software. The software can be downloaded from our website.

Getting Started:

To get started with NAP, just open the application. NAP is designed to import a network diagram from a .csv file. Select the “Choose File” button and locate the .csv file containing a network diagram on your computer (see “Formatting the .csv file” section). Alternatively, enter your data manually into our prompts, then select “Analyze” to determine the duration of a network diagram. You have the option to save the result as a .txt or .pdf file for later use. To analyze another file, select “Back,” which will then redirect you back to the application’s main page.

Formatting the .csv file:

NAP uses .csv files generated from programs like Excel to handle the network diagrams. Formatting this file is crucial to the program outputting the correct result. Each network diagram must contain a list of activities, predecessors, and durations. An activity with multiple predecessors must be separated by a semicolon, nothing else. An example file is shown below:

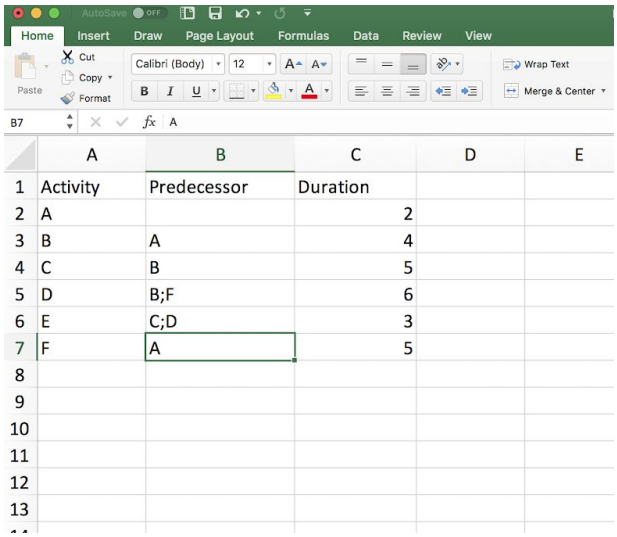


Figure 1: Screenshot that demonstrates proper formatting of network diagram using MS Excel

User Interface Overview:

Main Window:

The NAP user interface is designed to be intuitive and functional. The program's main window displays the option to choose .csv files or enter data manually for a network diagram. The "Analyze" button runs NAP's advanced algorithm where it will then display all paths on the network. Any error caused by the user will be displayed in red at the bottom of the window.

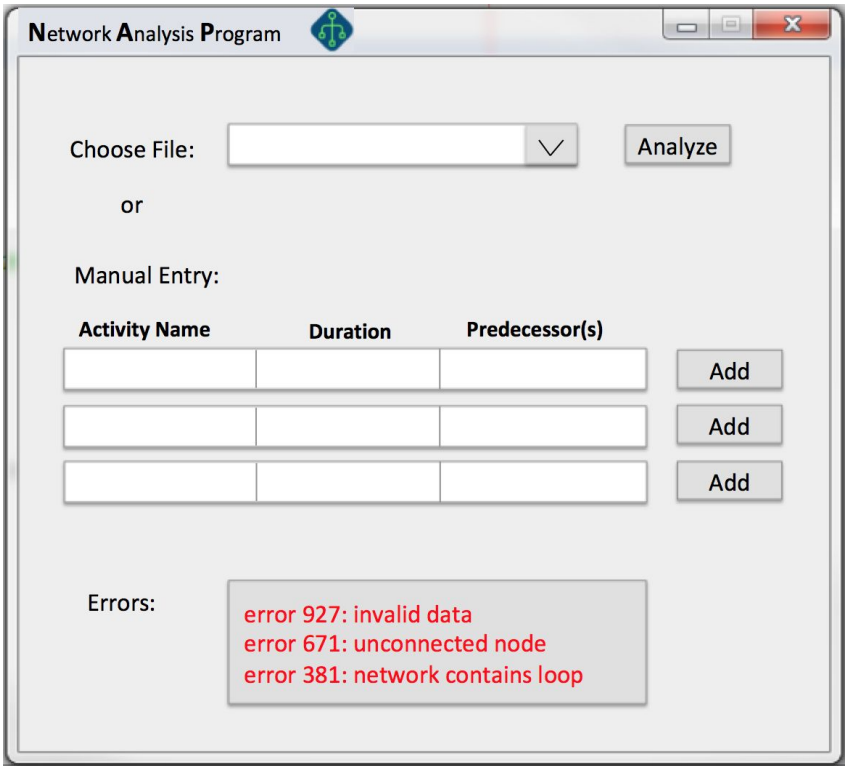


Figure 2: Example of how user will choose file/create network diagram, along with potential error message(s)

Results Window:

Results are displayed in descending order by the length of the path's duration. All network paths are displayed for the given network diagram. NAP allows the results to be saved as a .pdf or .txt file. See the figures below:

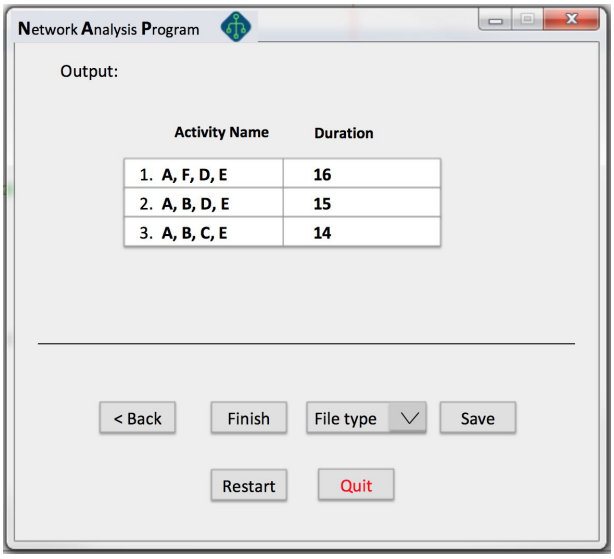


Figure 3: Demonstration of how paths and duration will be displayed

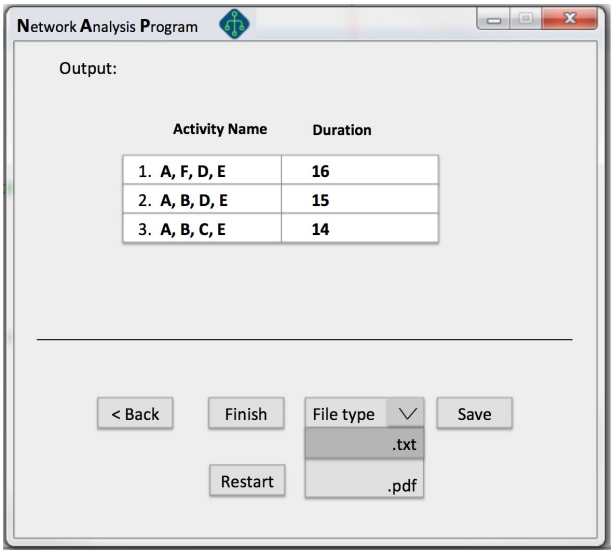


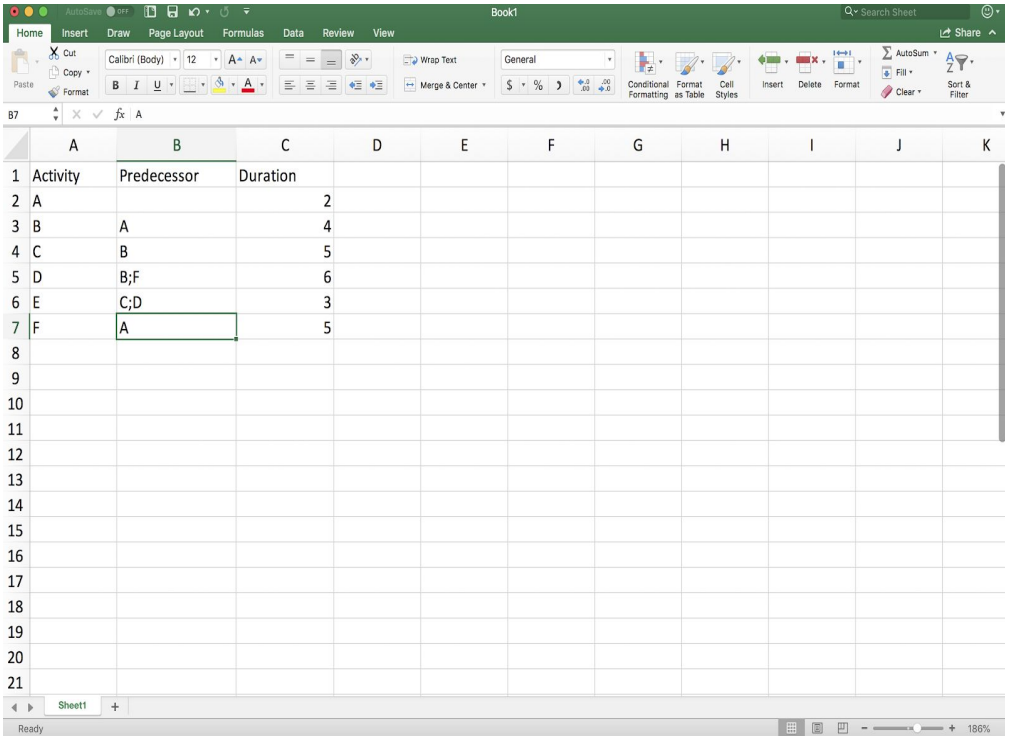
Figure 4: Demonstration of option to save file as .txt or .pdf file

Example runs:

Provided below are examples of a network diagram with proper formatting (Figure 5) and some that would produce errors (Figures 6-8). Utilize these to help identify any data or input errors.

Successful Run:

The input file below is formatted correctly. All activities have different names; all predecessors refer to activities that exist; and all durations are positive integers. Note: If an activity has multiple predecessors, then it must be separated by a semicolon. See Figure 5 below:



	A	B	C
1	Activity	Predecessor	Duration
2	A		2
3	B	A	4
4	C	B	5
5	D	B;F	6
6	E	C;D	3
7	F	A	5

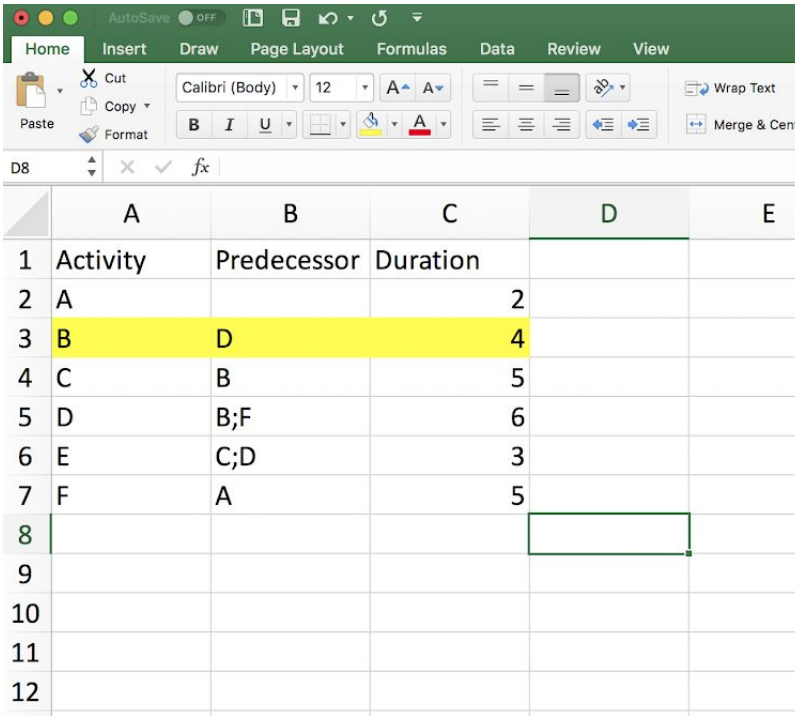
Figure 5: Example of a file with proper formatting

Error Conditions:

Listed below are common file errors that result in an invalid output.

Infinite Loop:

A loop occurs when an activity A has a predecessor B, which itself is dependent on A as a predecessor.

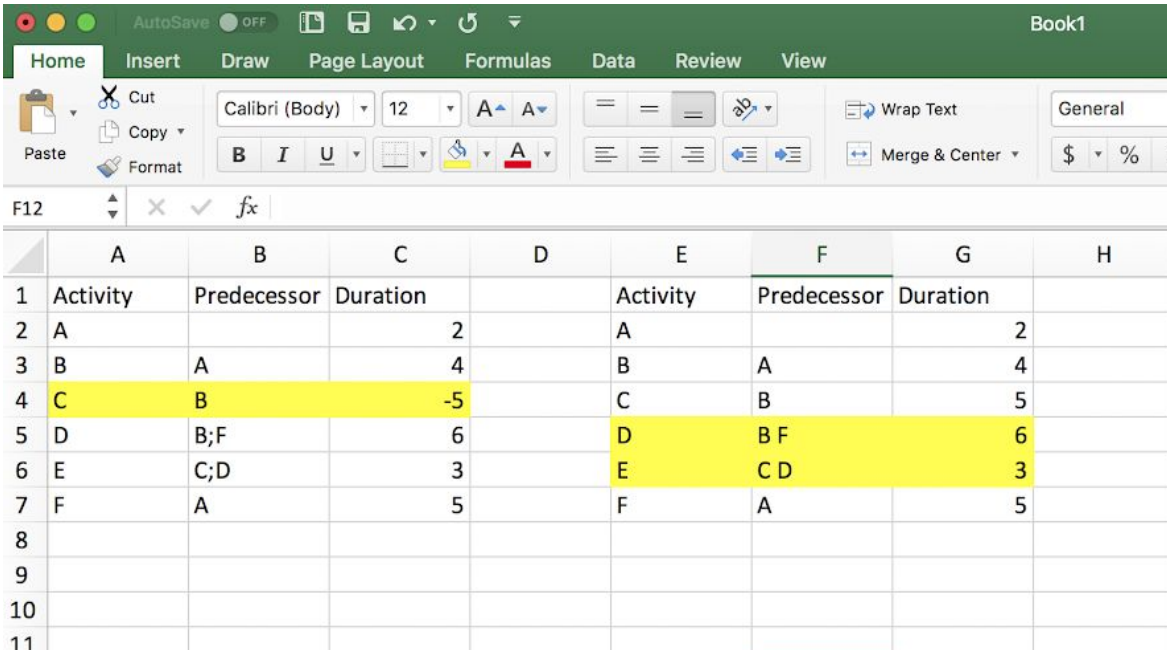


	A	B	C	D	E
1	Activity	Predecessor	Duration		
2	A		2		
3	B	D	4		
4	C	B	5		
5	D	B;F	6		
6	E	C;D	3		
7	F	A	5		
8					
9					
10					
11					
12					

Figure 6: Example of a file with invalid predecessor, resulting in infinite loop

Invalid Data:

Invalid data can include any durations that are not positive integers, activities without names, and predecessors that do not exist.

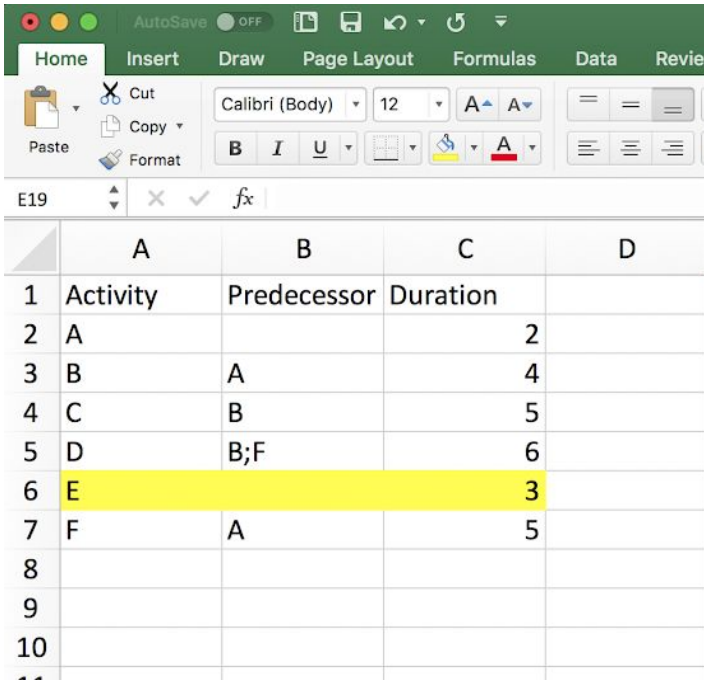


	A	B	C	D	E	F	G	H
1	Activity	Predecessor	Duration		Activity	Predecessor	Duration	
2	A		2		A		2	
3	B	A	4		B	A	4	
4	C	B	-5		C	B	5	
5	D	B;F	6		D	B F	6	
6	E	C;D	3		E	C D	3	
7	F	A	5		F	A	5	
8								
9								
10								
11								

Figure 7: Examples demonstrating invalid values for duration and predecessors, as well as missing semicolons separating predecessors

Unconnected Activity Error:

Any activity besides the start activity that has no predecessors, and any activity besides the end activity that is not a predecessor to another activity, is considered unconnected and renders the entire network invalid.



	A	B	C	D
1	Activity	Predecessor	Duration	
2	A		2	
3	B	A	4	
4	C	B	5	
5	D	B;F	6	
6	E		3	
7	F	A	5	
8				
9				
10				

Figure 8: Example of input file that results in loop due to lack of predecessor

Restarting:

To restart the program, simply press the button labeled “Restart”. Any existing input or output that has not been saved will be lost upon restarting.

Ending the program:

To end the program, either close the window running the program or click the “Quit” button. All output or output files must be saved before quitting or they will be lost.

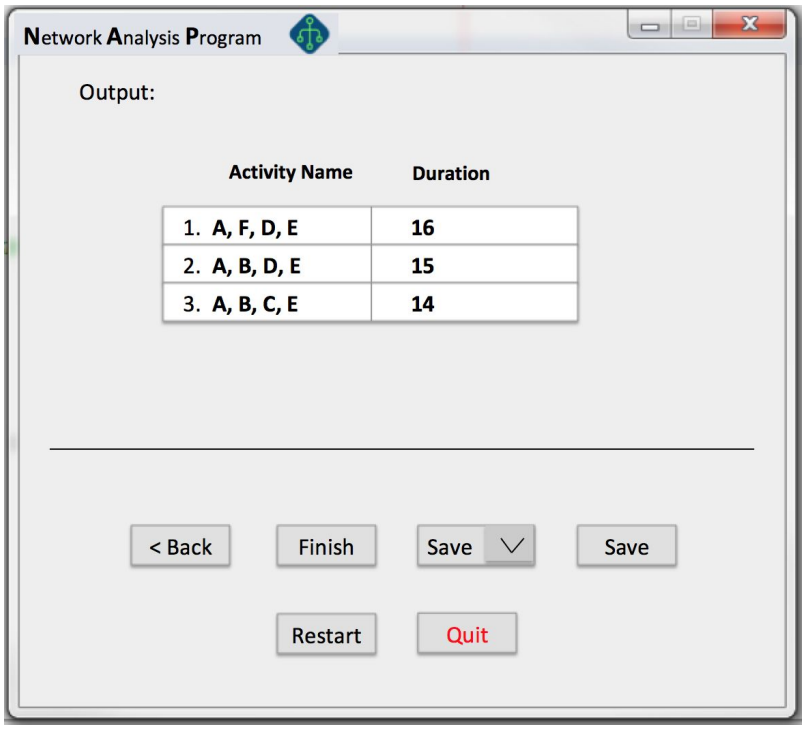


Figure 9: Demonstration of option buttons, specifically “Restart” and “Quit”