CSE360 SEMESTER PROJECT

PHASE 1 SUBMISSION: NETWORK PATH ANALYZER USER GUIDE

Ali Altunkaya Team #9

Team Members:
Dressman, Brandon
Sneberger, Michael
VanDiepenbos, Scott

Submitted via Blackboard:
by September 25, 2018

Contents

Introduction:

The Network Path Analyzer is a program designed to allow user to enter various project activities along with their durations and dependencies (activities which must occur prior) into the program which will then calculate the various work paths that could possibly flow through the project – particularly identifying the critical or shortest path.

Overview:

To use the program a user inputs their data (activities/durations/dependencies) in a presented table. An unlimited number of activities is allowed. If any data is entered incorrectly an error message is shown and re-entry requested. After running the program, output is presented in table format showing all various activity paths through the project, with the critical path being the first pathway shown. If the project as entered allows for a "cycle" or looping back upon itself, an error message will be shown as cycles are not permitted.

System Requirements and Installation:
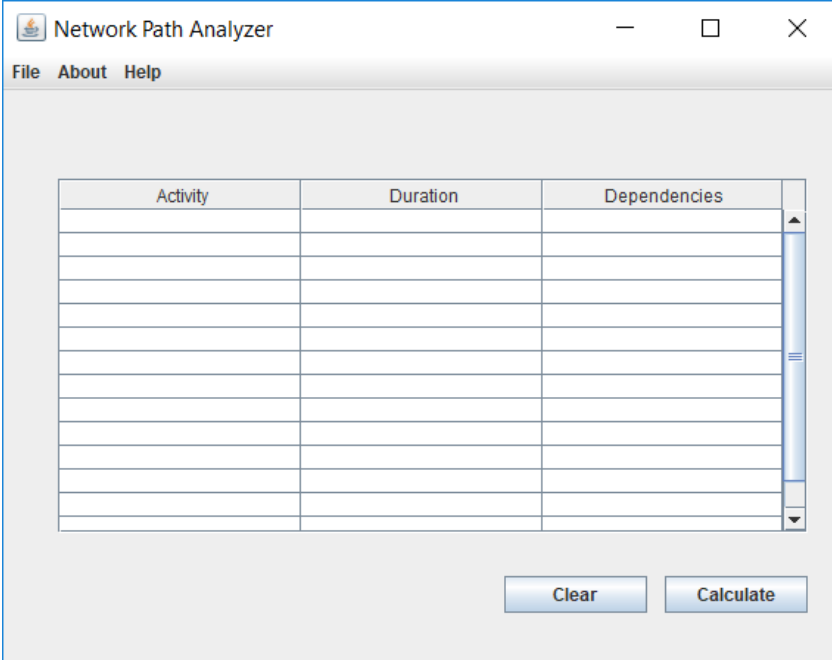
Requirements for running the program:

An active Java Runtime Environment (this is present in most modern computers) and the .exe file for the program.

Installation:

No formal installation is required. User merely double-clicks the .exe file to initiate the program.

Getting Started:

Upon opening the .exe file user will be presented with an entry table that appears as follows:

Input:

Once presented with the Entry Table, user will enter the various activities comprising the project by entering the following areas of information into the appropriate cell of the Entry Table:

- activity name as a string of characters;
- duration of the activity as an integer;
- list of dependencies (predecessors), *i.e.* what activity or activities must be completed prior to activity. Note that a listed dependency must be entered exactly as the activity was initially entered.

Data Entry Limitations:

Note that all activities must be connected to another activity, *i.e.* each activity must either serve as a predecessor or have a predecessor. The starting activity must not depend on any prior activity, and the final activity must not serve as a predecessor to any activity.

Maximum Number of Activities:

There is no maximum number of activities and/or dependencies(predecessors).

Error Detection:

If an error is detected in user's input, user will be notified and re-prompted for new input.

Running the Program:

Once user has completed their input, they click on the CALCULATE button on the lower right of window to begin processing.

Re-Running the Program:

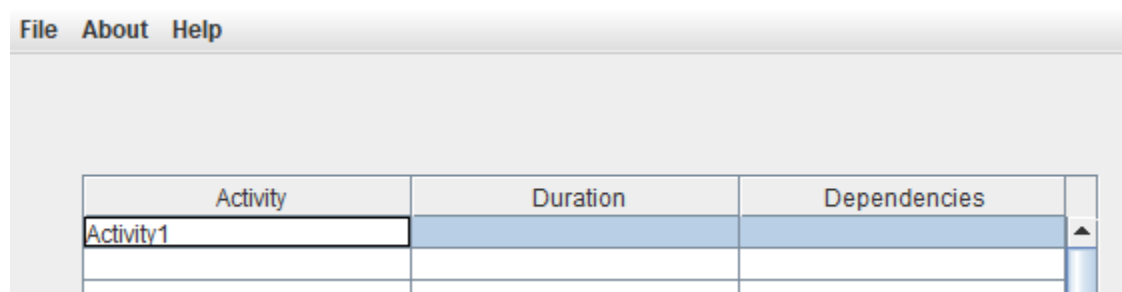After the program has run user has two options for re-running the program:

- user can retain and/or modify the currently entered data; or
- user can delete the currently entered by clicking the CLEAR button data and start anew.

Example Run of the Program Number 1:

Entering Path Information:

Start by entering the activity name of the first activity. (note: it is not required to start with the root node.)

Next, enter the duration value rounded up to the nearest integer.

Activity	Duration	Dependencies
Activity1	5	

Finally, enter any dependencies this activity will have. If more than one dependency, separate entries using commas. If no dependencies than Activity is assumed to be the root node. Only one root node is allowed.

Activity	Duration	Dependencies
Activity1	5	Activity2, Activity3

Calculating Paths/Output:

After entering all Activities with their corresponding Duration and Dependencies you are now ready to calculate all available paths. Click the calculate button at the bottom of the screen and the Program will output the following: a list of all possible activity paths through the project, with the critical or fastest path being listed first. Paths after the critical path are listed in descending order by duration.

Activity	Duration	Dependencies
A	5	
B	8	A
C	3	B
D	7	A
E	4	C, D

Clear   Calculate

All paths are displayed in descending order. Critical path will be the first path displayed on the list.

Path	Total Duration
(A,B,D,E)	19
(A,C,D,E)	17

Close

Example Run of the Program Number 2:

Entering Path Information:

Here is a how an alternative entry of data might appear:

Activity	Duration	Dependencies
A	3	
B	4	A
C	2	A
D	5	B
E	1	C
F	2	C
G	4	D,E
H	3	F,G

Clear   Calculate

Calculating Paths/Output:

After entering all Activities with their corresponding Duration and Dependencies as show above user is now ready to calculate all available paths. Click the calculate button at the bottom of the screen and the Program will output the following: a list of all possible activity paths through the

project, with the critical or fastest path being listed first. Paths after the critical path are listed in descending order by duration. Note that the critical path – or path with the longest overall duration is listed first:

Path	Total Duration
A,B,D,G,H	19
A,C,E,G,H	13
A,C,F,H	10

Close

Examples of How an Error Message Appears:

Error if User Mis-Enters Activity name:

As instructed, names of activities must be a letter of string of letters, not a number. If user mistakenly enters a number for an activity name an error like the following will be shown:

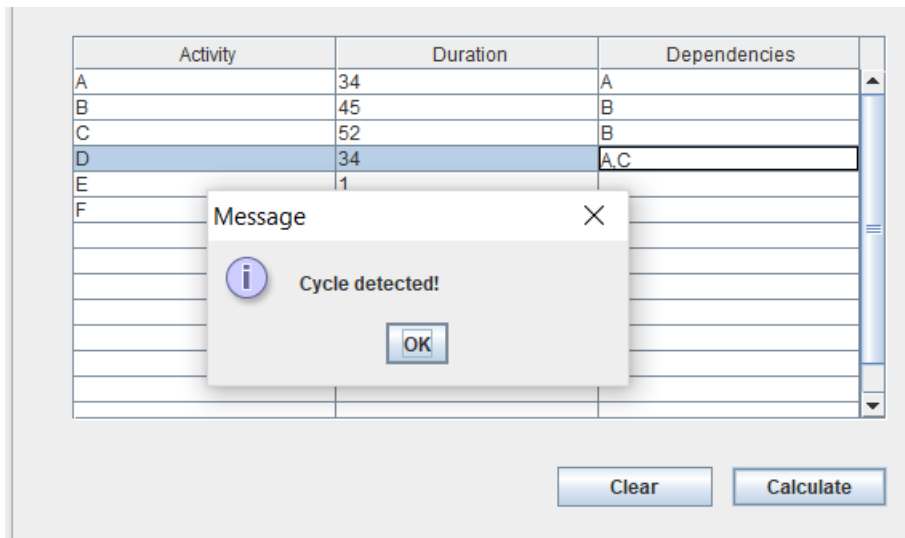| Activity	Duration	Dependencies
145		

Message ✕

ⓘ   **Activity input not valid**

OK

Clear   Calculate

Error if The Project as Entered Features a Cycle:

As instructed, cycles are not allowed in entered projects. If upon analysis a project is found to contain a cycle, an error like the following will be shown:

Conclusion:

The creators of the NETWORK PATH ANALYZER are pleased to provide users with an easy, efficient way to identify critical pathways in projects of all sizes. Users enter the various activities and the program performs the mathematical organization and calculation required to find the critical path through the project.