Hiking Trail Problem

The project requires use of yaml. The best practice for doing a project such as this is to use already created packages to implement the project. I normally develop under Linux. I presently do not have a Linux system but only have a Windows 10 system. I have used Windows 10 to develop this project, even though I do not consider myself to be a Windows 10 developer (but I am learning). My first step was to download and build a C++ yaml package for Windows 10. This is available on:  
<https://github.com/jbeder/yaml-cpp>  
Many thanks to Jesse Bender.

The next step is to discuss the algorithm. The goal is to cross a bridge as quickly as possible. A single torch is available that needs to accompany any hikers that cross the bridge. A maximum of two hikers can cross the bridge at once. It seems self-evident that the fastest hiker needs to accompany all the remaining hikers. So given the speed of a given hiker (excluding the fastest hiker) is:

And the speed of the fastest hiker is:

For a bridge of given length *l*, for all hikers, the time it will take to cross the bridge is:

The *n-1* is because the fastest hiker does not need to go back across the bridge after the last hiker is accompanied across.

Creating the yaml file for this problem:

* bridge: 1  
   length: 100  
   hikers:  
  - hiker: A

speed: 100  
- hiker: B  
 speed: 50  
- hiker: C  
 speed: 20  
- hiker: D  
 speed: 10

* bridge: 2  
   length: 250  
   hikers:  
  - hiker: E  
   speed: 2.5
* bridge: 3  
   length: 150  
   hikers:  
  - hiker: F  
   speed: 25  
  - hiker: G  
   speed: 15

Solving the problem by hand in this table as a check for the program:

|  |  |  |
| --- | --- | --- |
| **Bridge Length** | **Hikers Speed** | **Time** |
| 100 | A B 50 | 2 |
|  | A 100 | 1 |
|  | A C 20 | 5 |
|  | A 100 | 1 |
|  | A D 10 | 10 |
|  |  | Subtotal 19 |
| 250 | A B 50 | 5 |
|  | A 100 | 2.5 |
|  | A C 20 | 12.5 |
|  | A 100 | 2.5 |
|  | A D 10 | 25 |
|  | A 100 | 2.5 |
|  | A E 2.5 | 100 |
|  |  | Subtotal 150 |
| 150 | A B 50 | 3 |
|  | A 100 | 1.5 |
|  | A C 20 | 7.5 |
|  | A 100 | 1.5 |
|  | A D 10 | 15 |
|  | A 100 | 1.5 |
|  | A E 2.5 | 60 |
|  | A 100 | 1.5 |
|  | A F 25 | 6 |
|  | A 100 | 1.5 |
|  | A G 15 | 10 |
|  |  | Subtotal 109 |
|  |  | Total 278 |

Testing the program needs to include boundary conditions. This would be a single hiker traveling across a bridge as well as two hikers traveling across a bridge. I tested for these conditions by deleting all but the first hiker, then all but the first and second hikers in the yaml file.

Best practices include using existing packages and code when possible to implement a solution (the yaml package), commenting your code adequately, breaking code into paragraphs where the first line of each paragraph explains why you are writing this block of code. Comments need to answer why. They should never answer what – that is the purpose of the code itself.

See the comments in the code for explanations of what the code does.

I have a known problem with this code that I didn’t research. Invalid yaml files cause the program to crash. I would have to check more into jbender’s package to see if there were more diagnostics available.