ASSIGNMENT 1

Comp 4106

Max DeMelo – 100937368

BRIDGE AND TORCH

State Space

For the state space for the bridge and torch problem I used 2 array lists of integers which represented each side of the bridge, *peopleLoB* stood for people on the left hand side of the bridge (also known as people who haven't crossed) and *peopleRoB* for people that are on the right side of the bridge. I also had a Boolean called *torchLeft* which I used to determine which side of the bridge the torch was on. This allowed me to determine what moves were allowed because only someone with the torch can cross the bridge.   Finally, just to double check at the end I had the number of people at the beginning saved to confirm my program didn't lose or delete anyone. The state for this problem is considered complete when all the integers or people have been moved from the left side of the bridge to the right hand side, as well as the right hand side having the torch.

Heuristics

1: SUM OF PEOPLE STILL REMAINING ON LEFT SIDE OF THE BRIDGE

I calculated this heuristic value using the current cost of that particular node and the sum of the rest of the people (because they are just stored as a number) still on the left side of the bridge. This allows the AI to favor nodes that have people to take longer to cross on the other side.

2: BEST TRAVEL TIME REMAINING

I calculated this heuristic value using the current cost of that particular node and a rough estimation of the best possible time the rest of the children of this node could possibly have allowing the program to prune nodes that have unfavorable children.

3: AVERAGE BETWEEN BOTH

I calculated this heuristic value using the current cost of that particular node and the average of both previous calculations done on the same node. Allowing it to favor having more time consuming people on the right hand side of the bridge as well as the shortest path to get them there.

SPACE MANAGEMENT

State Space

The state space for the space management problem was divided into two parts, the values and the characteristics. First the board is represented by an array of integer arrays or a two dimensional array. The characteristics of the state are put in during construction of the original state, these characteristics include the width and height of the board, the number of blank spaces within the board and finally the number range that will be represented by this board (this number is calculated by width*height – blank spaces). On initial creation this board is randomized with all these factors mixed in. The state for this problem is considered complete when all the numbers are going increasing order around the outside of the grid and spiraling in on itself having all the blank spaces on the inside. Please see below for an example of a 4 by 4 grid with 1 blank space being complete.

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

Figure 1: Example of completed 4 by 4 grid

Heuristics

1: TILES STILL OUT OF PLACE

I calculated this heuristic value using the current depth of that particular node and the number of tiles that were still out of place. This allows the AI to focus nodes that have more of the grid filled in properly.

2: TOTAL REALTIVE DISTANCE

I calculated this heuristic value using the current depth of that particular node and the total relative distance of each number from its prospective "compete" coordinate. For example, in the above example if let's say the 4 was located at (1,2) instead of (3,0) then its relative distance would be 4. This would be done for all numbers in the grid and that total value is used as the total relative distance from the solution.

3: AVERAGE BETWEEN BOTH

I calculated this heuristic value using the current depth of that particular node and the average of both previous calculations done on the same node.