Name : Lee Tsang Keung

Student ID : 19062155D

Program Name : 19062155D_program

I. The problem

The problem is to bring Moses and seven other people from Egypt to Israel without having anyone beaten up in a minimum number of trips.There are several details about this problem:

1. The boat cannot be operated on its own. The boat can be operated by male only .
2. The maximum capacity of the boat is two.
3. At any time , Moses have to stay with Pharaoh
4. Ahab will beat up Ananias's wife and servant when Ananias's wife and servant without Ananias.
5. Ananias will beat up Abah's wife and servant when Abah's wife and servant without abah.

II. Data abstraction

System States The goal of this data abstraction step is to transform the human-readable problem to a machine-readable form. For convenience, we identify the three couples by order in list . We define the state of this system by a 9-tuple of Boolean values—(E)ast and (W)est—which indicates the current locations of the boat and the eight people. There are a total of 2^9(512) states. The nine Boolean values are ordered in this way: the boat, Moses, Pharaoh and two families: (1) Ahab (husband), Jezebel (wife) and a servant and (2) Ananias (husband), Sapphira (wife) and a servant. For example, (E,E,E,W,W,W,W,W,W) is for the boat , Moses and Pharaoh on the East side, and the other two family on the West side.

Nodes There are several illegal case for the node .As a result , part a in program can find out all the illegal case .Those legal node can be used to generate the shortest path.

Links Each legal node can be linked when they fulfill the following requirement

1. The position of boat change
2. Except the boat , one to two person have to be change position .

Since the legal node is generate in part a , all linking must be legal . If the path is generated starting with "EEEEEEEEE" and end with "WWWWWWWWW" it is a valid path.

A shortest-path problem By putting the nodes and links together, we have several valid path. Therefore,We have to find the short path thought comparing the length of path .When a path is generated , its number of step will be recorded . If the new path is shorter than the previous path , it will be record as the shortest path . After all path is generated , we can get the path which is shortest. The shortest path have 17 steps Which shown as below:

(1) The Moses and Pharaoh go from the east to the west.

(2) The Moses goes from the west to the east.

(3) The Moses and servant of Ananias go from the east to the west.

(4) The Moses and Pharaoh go from the west to the east.

(5) The Ananics and Sapphira go from the east to the west.

(6) The Ananics goes from the west to the east.

(7) The Ahab and Ananics go from the east to the west.

(8) The Ahab goes from the west to the east.

(9) The Moses and Pharaoh go from the east to the west.

(10) The Ananics goes from the west to the east.

(11) The Ahab and Ananics go from the east to the west.

(12) The Ahab goes from the west to the east.

(13) The Ahab and servant of Ahab go from the east to the west.

(14) The Moses and Pharaoh go from the west to the east.

(15) The Moses and Jezebel go from the east to the west.

(16) The Moses goes from the west to the east.

(17) The Moses and Pharaoh go from the east to the west.

Abstract data types The above data processing require difference data type to help generate the output

1. A Boolean value for the location
2. A Sets of illegal states is use for generate a Sets of legal states
3. A graph for the relationship among all the legal states
4. A list of legal states as a solution to the problem

IV. Program design

The goal of this section is to design a modular program by breaking the program into a set of functions. Following the best practice, each function will perform only a single task. Table below shown that each small program detail.

Function	Inputs	Outputs	Comments
Alllegalstate()	Null	A list legal state	
Solver()	Null	null	Generate all output by human friendly graph.
genGraph()	s	A graph of all legal states	Generate a graph of all legal states
neighborNode()	states s1, s2	True or false	Check whether s1 and s2 are neighbors
genShortestPath()	G, (E,E,E,E,E,E,E,E,E), (W,W,W,W,W,W,W,W,W), an empty path	a solution path	Generate a shortest path . Generate possible path and record in global usefulpath

printPath()	P	Null	Print a human-readable solution