Brute force using depth first traversal

Hug left wall

Algorithmic using breadth first traversal

Find node closest to end for next move

Struct node{

String type; //intersection, start, finish, edge/deadend;

Node\* north; //node north of current (null if no white space in that direction)

Node\* south;

Node\* east;

Node\* west;

Int northDistance; //units between current and north node (null if correlated node is null)

Int southDistance;

Int eastDistance;

Int westDistance;};

Insert start node pointer to inread start node

Int\* fastestPath;

Int fastestDistance;

Node \*Current=start;

Begin traversal;

While(current!=start){

Traverse;

Build path with N=1, E=2, S=3, W=4;

If(finish){

tempDistance=findDistance(int\* path, node\* start);

if(tempDistance<fastestDistance){

fastestDistance=tempDistance;

path = fastestPath;

}

}

Else If(edge) backtrack;

Else if(no options) assign as deadend and backtrack;

}

Int findDistance(int\* path, node\* start){

Sum=0;

Node\* curr = start;

Switch(direction){

Case1: sum+=distance from curr to north node, curr=north node

Case2: to east node; = east node

Case3: to south node; = south node

Case4: to west node; = west node

}

Return sum;

}