323.35 Project 8: 8 Puzzle A* CPP

Student: Matthew Flammia
Project Due Date: 11/28/2020

```
*******************************
IV. main () // A* algorithms
*******************************
Step 0: initialConfiguration  get from inFile1
goalConfiguration  get from inFile2
startNode  create a AstarNode for startNode with initialConfiguration
goalNode  create a AstarNode for goalNode with goalConfiguration
OpenList  create a linked list with a dummy node
CloseList  create a linked list with a dummy node
Step 1: startNode's gStar  0
startNode's hStar  computeMissTiles (StartNode)
startNode's fStar  startNode's gStar + startNode's hStar
listInsert (startNode) // Insert startNode into OpenList, in ascending order w.r.t. fStar
Step 2: currentNode  remove (OpenList)
Step 3: if (isGoalNode (currentNode))// A solution is found!
printSolution (node, outFile2)
return or exit the program
Step 4: childList  constructChildList (currentNode)
Step 5: child  pop (childList)
Step 6: child's gStar  computeGstar (child )
child's hStar  computeHstar (child)
child's fStar  child's gStar + child's hStar
Step 7: if child is not in OpenList and not in CloseList
 Insert child into OpenList
 child's parent  currentNode // back pointer
 else if child is in OpenList and child's f* is better (<) than the old node's f* in OpenList
replace child with the old child in OpenList,
//i.e., do a delete and an insert
child's parent  currentNode // back pointer
 else if child is in CloseList and its f* is better (<) than the f* of old node on CloseList
remove child from CloseList
Insert child into OpenList
child's parent  currentNode // back pointer
Step 8: repeat Step 5 to Step 7 until childList is empty
Step 9: Print "This is Open list:" to outFile1
 printList (OpenList, outFile1)
Print "This is CLOSE list:" to outFile1
printList (CloseList, outFile1)
Print up to 20 loops!
```

Step 10: repeat step 2 to step 9 until currentNode is a goal node or OpenList is empty.
Step 11: if OpenList is empty but currentNode is NOT a goal node,
print error message: "no solution can be found in the search!" to outFile1
Step 12: close all files

<u>Source code</u>

```
#include <iostream>
#include <stdlib.h>
#include <fstream>
using namespace std;

class AstarNode{
        public:
        int configuration[9];
        int gStar;
        int hStar;
        int fStar;
        AstarNode* parent;
        AstarNode* next;

        AstarNode(){
                for(int i=0;i<9;i++)
                        this->configuration[i] = 0;
                this->gStar = 0;
                this->hStar = 0;
                this->fStar = 0;
                this->parent = nullptr;
                this->next = nullptr;
        }

        void printNode(ofstream& output){
                output<<"< ";
                for(int i=0;i<9;i++){
                        output<<this->configuration[i]<<" ";
                }
                output<<"| "<<this->fStar<<" | ";
                if(this->parent == nullptr){
                        output<<"NULL >";
                }
                else{
                        for(int i=0;i<9;i++){
                                output<<this->parent->configuration[i]<<" ";
                        }
                        output<<">";
                }
                output<<endl;
        }
};

class AStarSearch{
        public:
        AstarNode startNode;
        AstarNode goalNode;
        AstarNode* openList;
        AstarNode* closeList;
        AstarNode* childList;
        int h2array[9][9] = {{0,1,2,1,2,3,2,3,4},
                                                  {1,0,1,2,1,2,3,2,3},
                                                  {2,1,0,3,2,1,4,3,2},
                                                  {1,2,3,0,1,2,1,2,3},
                                                  {2,1,2,1,0,1,2,1,2},
                                                  {3,2,1,2,1,0,3,2,1},
                                                  {2,3,4,1,2,3,0,1,2},
                                                  {3,2,3,2,1,2,1,0,1},
                                                  {4,3,2,3,2,1,2,1,0}};

        AStarSearch(){
                this->openList = new AstarNode();
                this->closeList = new AstarNode();
                this->childList = nullptr;
        }
        //simple methods
        int computeGstar(AstarNode* node){
                int temp;
                temp = node->parent->gStar + 1;
                return temp;
        }
        int computeHstar(AstarNode* node){
                int miss = 0;
                for(int i=0; i<9;i++){
                        miss +=
this->h2array[node->configuration[i]][this->goalNode.configuration[i]];
                }
                return miss;
        }
        bool match(int config1[], int config2[]){
                for(int i=0;i<9;i++){
                        if(config1[i] != config2[i])
```

```
                    return false;
                }
                return true;
        }
        bool isGoalNode(AstarNode* node){
                return this->match(node->configuration, this->goalNode.configuration);
        }
        //complex methods
        bool checkAncestors(AstarNode* currentNode, AstarNode* parent){
                if(parent == nullptr){
                        return false;
                }
                else if(this->match(currentNode->configuration, parent->configuration)){
                        return true;
                }
                else{
                        return this->checkAncestors(currentNode, parent->parent);
                }
        }
        AstarNode* constructChildList(AstarNode* currentNode, ofstream& output){
                AstarNode* dummy = new AstarNode();
                for(int i=0;i<9;i++){
                        if(currentNode->configuration[i] == 0){
                                if(i+1 < 9){
                                        int temp = 0;
                                        AstarNode* newNode = new AstarNode();
                                        //configuration copy
                                        for(int j=0;j<9;j++)
                                                newNode->configuration[j] =
    currentNode->configuration[j];
                                        //swap
                                        temp = newNode->configuration[i];
                                        newNode->configuration[i] = newNode->configuration[i+1];
                                        newNode->configuration[i+1] = temp;
                                        //g h f star assignment
                                        newNode->parent = currentNode;
                                        newNode->gStar = this->computeGstar(newNode);
                                        newNode->hStar = this->computeHstar(newNode);
                                        newNode->fStar = newNode->gStar + newNode->hStar;
                                        //output<<"new +1 node:\n";
                                        //newNode->printNode(output);
                                        if(!this->checkAncestors(newNode,currentNode)){
                                                this->push(newNode, &dummy);
                                        }
				else{
					//output<<"failed to insert\n";
					delete newNode;
				}
			}
			if(i+3 < 9){
				int temp = 0;
				AstarNode* newNode = new AstarNode();
				//configuration copy
				for(int j=0;j<9;j++)
					newNode->configuration[j] =
currentNode->configuration[j];
				//swap
				temp = newNode->configuration[i];
				newNode->configuration[i] = newNode->configuration[i+3];
				newNode->configuration[i+3] = temp;
				//g h f star assignment
				newNode->parent = currentNode;
				newNode->gStar = this->computeGstar(newNode);
				newNode->hStar = this->computeHstar(newNode);
				newNode->fStar = newNode->gStar + newNode->hStar;
				//output<<"new +3 node:\n";
				//newNode->printNode(output);
				if(!this->checkAncestors(newNode,currentNode)){
					this->push(newNode, &dummy);
				}
				else{
					//output<<"failed to insert\n";
					delete newNode;
				}
			}
			if(i-1 >= 0){
				int temp = 0;
				AstarNode* newNode = new AstarNode();
				//configuration copy
				for(int j=0;j<9;j++)
					newNode->configuration[j] =
currentNode->configuration[j];
				//swap
				temp = newNode->configuration[i];
				newNode->configuration[i] = newNode->configuration[i-1];
				newNode->configuration[i-1] = temp;
				//g h f star assignment
				newNode->parent = currentNode;
```

```
                    newNode->gStar = this->computeGstar(newNode);
                    newNode->hStar = this->computeHstar(newNode);
                    newNode->fStar = newNode->gStar + newNode->hStar;
                    //output<<"new -1 node:\n";
                    //newNode->printNode(output);
                    if(!this->checkAncestors(newNode,currentNode)){
                        this->push(newNode, &dummy);
                    }
                    else{
                        //output<<"failed to insert\n";
                        delete newNode;
                    }
                }
                if(i-3 >= 0){
                    int temp = 0;
                    AstarNode* newNode = new AstarNode();
                    //configuration copy
                    for(int j=0;j<9;j++)
                        newNode->configuration[j] =
currentNode->configuration[j];
                    //swap
                    temp = newNode->configuration[i];
                    newNode->configuration[i] = newNode->configuration[i-3];
                    newNode->configuration[i-3] = temp;
                    //g h f star assignment
                    newNode->parent = currentNode;
                    newNode->gStar = this->computeGstar(newNode);
                    newNode->hStar = this->computeHstar(newNode);
                    newNode->fStar = newNode->gStar + newNode->hStar;
                    //output<<"new -3 node:\n";
                    //newNode->printNode(output);
                    if(!this->checkAncestors(newNode,currentNode)){
                        this->push(newNode, &dummy);
                    }
                    else{
                        //output<<"failed to insert\n";
                        delete newNode;
                    }
                }
                return dummy;
            }
        }
        return nullptr;
    }
```

```
void listInsert(AstarNode* node, AstarNode** list){
        AstarNode* temp = *list;
        if(temp->next == nullptr){
                temp->next = node;
        }
        else{
                while(temp->next != nullptr){
                        if(temp->next->fStar > node->fStar)
                                break;
                        temp = temp->next;
                }
                if(temp->next == nullptr){
                        temp->next = node;
                }
                else{
                        node->next = temp->next;
                        temp->next = node;
                }
        }
}
AstarNode* listRemove(AstarNode** list){
        AstarNode* listhead= *list;
        if(listhead->next == nullptr){
                cout<<"ERROR! TRIED REMOVING FROM EMPTY LIST.\n";
                exit(-1);
        }
        AstarNode* temp = listhead->next;
        listhead->next = listhead->next->next;
        temp->next = nullptr;
        return temp;
}
//printing methods
void printList(AstarNode** list, ofstream& outfile1){
        AstarNode* temp = *list;
        while(temp != nullptr){
                temp->printNode(outfile1);
                temp = temp->next;
        }
        outfile1<<"~~~~~~~~~~~~~\n";
}
void printSolution(AstarNode* currentNode, ofstream& outfile2){
        cout<<endl<<"Solution found. Please see results file."<<endl;
        outfile2<<"Solution found.\nGoal\n";
        printSolutionHelp(currentNode, outfile2);
```

```
}
bool printSolutionHelp(AstarNode* node, ofstream& outfile2){
	if(node == nullptr){
		outfile2<<"Start Node\n";
		return true;
	}
	else{
		for(int i=0;i<9;i++){
			if(i==3 || i==6)
				outfile2<<endl;
				outfile2<<node->configuration[i]<<" ";
		}
		outfile2<<endl<<endl;
	}
	return printSolutionHelp(node->parent, outfile2);
}
//helper methods not on specs
void removeNode(AstarNode* node, AstarNode** list){
	AstarNode* listhead = *list;
	if(listhead->next == nullptr){
		return;
	}
	listhead = listhead->next;
	while(listhead->next != nullptr){
		if(match(node->configuration, listhead->next->configuration)){
			AstarNode* bye = listhead->next;
			listhead->next = listhead->next->next;
			delete bye;
			return;
		}
		listhead = listhead->next;
	}
	return;
}
void push(AstarNode* node, AstarNode** list){
	AstarNode* temp = *list;
	if(temp->next == nullptr){
		temp->next = node;
	}
	else{
		node->next = temp->next;
		temp->next = node;
	}
}
```

```
	bool inList(AstarNode* node, AstarNode** list){
		AstarNode* listhead = *list;
		if(listhead->next == nullptr){
			return false;
		}
		listhead = listhead->next;
		while(listhead != nullptr){
			if(match(node->configuration, listhead->configuration )&& node->fStar <
listhead->fStar){
				return true;
			}
			else if(match(node->configuration, listhead->configuration) &&
!(node->fStar < listhead->fStar)){
				return false;
			}
			listhead = listhead->next;
		}
		return false;
	}
};

int main(int argc, char* argv[]){
	//checks that correct args were supplied
	if(argc != 5){
		cout<<"Must have 4 arguments in this command to run correctly.\ninFile1,
inFile2, Debug, Results\n";
		return -1;
	}
	//creates input stream and checks that its readable
	ifstream inFile1(argv[1]);
	ifstream inFile2(argv[2]);
	if(!inFile1.good() || !inFile2.good()){
		cout<<"Failed to read input file, was name typed correctly?\n";
		return -1;
	}
	//output streams
	ofstream debug(argv[3]);
	ofstream results(argv[4]);
	//create configurations
	int initConfig[9];
	int goalConfig[9];
	//step 0
	AStarSearch AStar;
	for(int i=0;i<9;i++){
        inFile1 >> AStar.startNode.configuration[i];
        inFile2 >> AStar.goalNode.configuration[i];
}
inFile1.close();
inFile2.close();
//step 1
AStar.startNode.gStar = 0;
AStar.startNode.hStar = AStar.computeHstar(&AStar.startNode);
AStar.startNode.fStar = AStar.startNode.hStar;
AStar.listInsert(&AStar.startNode, &AStar.openList);
/**debug code
debug<<"Printing Goal:\n";
AStar.goalNode.printNode(debug);
debug<<"Debugging after inserting startnode:\n";
AStar.startNode.printNode(debug);
AStar.printList(&AStar.openList, debug);
**/
AstarNode* currentNode;
int counter = 0;
//step 10 loop
do{
        //step 2
        currentNode = AStar.listRemove(&AStar.openList);
        AStar.listInsert(currentNode, &AStar.closeList);
        /**
        debug<<"Loop number:"<<counter<<endl;
        debug<<"Outputting current node:\n";
        currentNode->printNode(debug);
        **/
        //step 3
        if(AStar.isGoalNode(currentNode)){
                AStar.printSolution(currentNode, results);
                return 0;
        }
        //step 4
        AStar.childList = AStar.constructChildList(currentNode, debug);
        //debug<<"Debugging after creating Child List:\n";
        //AStar.printList(&AStar.childList, debug);

        //step 8 loop
        while(AStar.childList->next != nullptr){
                //step 5
                AstarNode* child = AStar.listRemove(&AStar.childList);
                //step 6
```

```
                    child->gStar = AStar.computeGstar(child);
                    child->hStar = AStar.computeHstar(child);
                    child->fStar = child->gStar + child->hStar;

                    //debug<<"printing child node:\n";
                    //child->printNode(debug);

                    //step 7
                    bool inOpen = AStar.inList(child, &AStar.openList);
                    bool inClose = AStar.inList(child, &AStar.closeList);
                    if(!inOpen && !inClose){
                            //debug<<"not in either list\n";
                            AStar.listInsert(child, &AStar.openList);
                            child->parent = currentNode;
                    }
                    else if(inOpen){
                            //debug<<"in open and better f\n";
                            AStar.removeNode(child, &AStar.openList);
                            AStar.listInsert(child,&AStar.openList);
                            child->parent = currentNode;
                    }
                    else if(inClose){
                            //debug<<"in closed and better f\n";
                            AStar.removeNode(child, &AStar.closeList);
                            AStar.listInsert(child, &AStar.openList);
                            child->parent = currentNode;
                    }
                    else{
                            //debug<<"no where to put, deleting\n";
                            delete child;
                    }
            }
            //step 9
            if(counter < 20){
                    debug<<"This is Open List:"<<endl;
                    AStar.printList(&AStar.openList,debug);
                    debug<<"This is Close List:"<<endl;
                    AStar.printList(&AStar.closeList, debug);
            }
            cout<<"Current Loops:"<<++counter<<"\r";
    }while(!AStar.match(currentNode->configuration, AStar.goalNode.configuration) ||
AStar.openList->next == nullptr);
    //step 11
```

```
	if(AStar.openList->next == nullptr && !AStar.match(currentNode->configuration,
AStar.goalNode.configuration)){
		debug<<"ERROR! OPEN LIST EMPTY WITHOUT GOAL BEING
FOUND"<<endl;
		return -1;
	}
	//step 12
	debug.close();
	results.close();
	return 0;
}
```

<u>Outfile 1 (debug) first pair</u>

```
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
```

```
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
~~~~~~~~~~~~~
```

```
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
~~~~~~~~~~~~~
```

```
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
```

```
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
```

```
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
< 8 1 3 0 2 4 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 6 4 7 0 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 4 0 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 7 0 1 4 6 5 | 11 | 2 8 3 7 1 0 4 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 8 0 7 1 3 4 6 5 | 13 | 2 8 3 7 1 0 4 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 1 5 4 6 0 | 15 | 2 8 3 7 1 0 4 6 5 >
< 8 1 3 0 2 4 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 6 4 7 0 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 4 0 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
~~~~~~~~~~~~~~
```

```
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 8 0 7 1 3 4 6 5 | 13 | 2 8 3 7 1 0 4 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 1 5 4 6 0 | 15 | 2 8 3 7 1 0 4 6 5 >
< 8 1 3 0 2 4 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 6 4 7 0 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 4 0 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 2 0 3 7 8 1 4 6 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 7 6 1 4 0 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 0 7 1 4 6 5 | 18 | 2 8 3 7 0 1 4 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
```

```
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 0 1 4 6 5 | 11 | 2 8 3 7 1 0 4 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 8 0 7 1 3 4 6 5 | 13 | 2 8 3 7 1 0 4 6 5 >
< 2 8 3 7 0 4 6 1 5 | 13 | 2 8 3 7 1 4 6 0 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 1 5 4 6 0 | 15 | 2 8 3 7 1 0 4 6 5 >
< 8 1 3 0 2 4 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 6 4 7 0 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 4 0 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 2 0 3 7 8 1 4 6 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 7 6 1 4 0 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 7 1 4 6 5 0 | 17 | 2 8 3 7 1 4 6 0 5 >
< 2 8 3 0 7 1 4 6 5 | 18 | 2 8 3 7 0 1 4 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 0 1 4 6 5 | 11 | 2 8 3 7 1 0 4 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
```

```
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 2 3 8 0 4 7 6 5 | 5 | 1 2 3 0 8 4 7 6 5 >
< 2 8 3 1 6 4 0 7 5 | 13 | 2 8 3 1 6 4 7 0 5 >
< 2 8 0 1 4 3 7 6 5 | 13 | 2 8 3 1 4 0 7 6 5 >
< 2 8 0 7 1 3 4 6 5 | 13 | 2 8 3 7 1 0 4 6 5 >
< 2 8 3 7 0 4 6 1 5 | 13 | 2 8 3 7 1 4 6 0 5 >
< 1 2 0 3 8 4 7 6 5 | 13 | 1 2 3 0 8 4 7 6 5 >
< 1 2 3 7 8 4 0 6 5 | 13 | 1 2 3 0 8 4 7 6 5 >
< 2 0 8 3 1 4 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 4 3 1 0 7 6 5 | 14 | 2 8 0 3 1 4 7 6 5 >
< 2 8 3 1 6 4 7 5 0 | 15 | 2 8 3 1 6 4 7 0 5 >
< 2 3 0 1 8 4 7 6 5 | 15 | 2 0 3 1 8 4 7 6 5 >
< 2 8 3 1 4 5 7 6 0 | 15 | 2 8 3 1 4 0 7 6 5 >
< 2 8 3 1 4 7 0 6 5 | 15 | 2 8 3 1 4 0 7 6 5 >
< 8 3 0 2 1 4 7 6 5 | 15 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 1 5 4 6 0 | 15 | 2 8 3 7 1 0 4 6 5 >
< 8 1 3 0 2 4 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 6 4 7 0 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 8 1 3 2 4 0 7 6 5 | 16 | 8 1 3 2 0 4 7 6 5 >
< 2 0 3 7 8 1 4 6 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 7 6 1 4 0 5 | 16 | 2 8 3 7 0 1 4 6 5 >
< 2 8 3 7 1 4 6 5 0 | 17 | 2 8 3 7 1 4 6 0 5 >
< 2 8 3 0 7 1 4 6 5 | 18 | 2 8 3 7 0 1 4 6 5 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 2 8 3 1 0 4 7 6 5 | 7 | 2 8 3 1 6 4 7 0 5 >
< 0 8 3 2 1 4 7 6 5 | 9 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 1 6 4 7 0 5 | 10 | NULL >
< 2 8 3 0 1 4 7 6 5 | 10 | 2 8 3 1 0 4 7 6 5 >
< 2 8 0 3 1 4 7 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 2 8 3 7 1 4 0 6 5 | 11 | 2 8 3 0 1 4 7 6 5 >
< 0 2 3 1 8 4 7 6 5 | 11 | 2 0 3 1 8 4 7 6 5 >
< 8 1 3 2 0 4 7 6 5 | 11 | 8 0 3 2 1 4 7 6 5 >
< 2 8 3 7 0 1 4 6 5 | 11 | 2 8 3 7 1 0 4 6 5 >
< 2 0 3 1 8 4 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 2 8 3 1 4 0 7 6 5 | 12 | 2 8 3 1 0 4 7 6 5 >
< 8 0 3 2 1 4 7 6 5 | 12 | 0 8 3 2 1 4 7 6 5 >
```

```
< 2 8 3 7 1 0 4 6 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 2 8 3 7 1 4 6 0 5 | 12 | 2 8 3 7 1 4 0 6 5 >
< 1 2 3 0 8 4 7 6 5 | 12 | 0 2 3 1 8 4 7 6 5 >
~~~~~~~~~~~~~
```

Outfile 2 (results) first pair

```
Solution found.
Goal
1 2 3
8 0 4
7 6 5

1 2 3
0 8 4
7 6 5

0 2 3
1 8 4
7 6 5

2 0 3
1 8 4
7 6 5

2 8 3
1 0 4
7 6 5

2 8 3
1 6 4
7 0 5

Start Node
```

Outfile 1 (debug) second pair

```
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
~~~~~~~~~~~~~
This is Close List:
```

```
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
```

```
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
```

```
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
```

```
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
```

```
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
```

```
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
```

```
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
```

```
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
```

```
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
< 2 1 4 0 5 7 3 6 8 | 21 | 0 1 4 2 5 7 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
```

```
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
< 2 1 4 0 5 7 3 6 8 | 21 | 0 1 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 6 0 8 | 21 | 1 4 2 3 5 7 0 6 8 >
< 1 4 2 3 5 0 7 6 8 | 23 | 1 4 2 3 5 7 0 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
```

```
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
< 2 1 4 0 5 7 3 6 8 | 21 | 0 1 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 6 0 8 | 21 | 1 4 2 3 5 7 0 6 8 >
< 4 2 1 0 5 7 3 6 8 | 21 | 4 2 0 1 5 7 3 6 8 >
< 1 4 2 3 5 0 7 6 8 | 23 | 1 4 2 3 5 7 0 6 8 >
< 4 2 7 1 5 0 3 6 8 | 25 | 4 2 0 1 5 7 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
```

```
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 4 0 3 1 7 5 2 6 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 4 7 3 0 1 5 2 6 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 4 7 3 1 6 5 2 0 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
< 2 1 4 0 5 7 3 6 8 | 21 | 0 1 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 6 0 8 | 21 | 1 4 2 3 5 7 0 6 8 >
< 4 2 1 0 5 7 3 6 8 | 21 | 4 2 0 1 5 7 3 6 8 >
< 1 4 2 3 5 0 7 6 8 | 23 | 1 4 2 3 5 7 0 6 8 >
< 4 2 7 1 5 0 3 6 8 | 25 | 4 2 0 1 5 7 3 6 8 >
~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
```

```
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
This is Open List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 7 2 5 8 3 0 6 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 0 4 7 5 3 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 1 4 3 7 5 0 2 6 8 | 19 | 1 4 0 7 5 3 2 6 8 >
< 4 0 3 1 7 5 2 6 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 4 7 3 0 1 5 2 6 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 4 7 3 1 6 5 2 0 8 | 19 | 4 7 3 1 0 5 2 6 8 >
< 1 4 7 2 0 5 3 6 8 | 20 | 1 4 7 2 5 0 3 6 8 >
< 1 5 4 2 0 7 3 6 8 | 20 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 5 0 7 3 6 8 | 20 | 1 4 2 0 5 7 3 6 8 >
< 4 5 2 1 0 7 3 6 8 | 20 | 4 0 2 1 5 7 3 6 8 >
< 1 4 7 5 0 3 2 6 8 | 20 | 1 4 7 0 5 3 2 6 8 >
< 4 5 7 1 0 3 2 6 8 | 20 | 4 0 7 1 5 3 2 6 8 >
< 4 7 3 1 5 8 2 6 0 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 3 1 5 2 0 6 8 | 20 | 4 7 3 1 5 0 2 6 8 >
< 4 7 1 2 5 3 0 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 4 7 1 5 0 3 2 6 8 | 20 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 3 6 0 8 | 21 | 1 4 7 2 5 3 0 6 8 >
< 2 1 4 0 5 7 3 6 8 | 21 | 0 1 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 6 0 8 | 21 | 1 4 2 3 5 7 0 6 8 >
< 4 2 1 0 5 7 3 6 8 | 21 | 4 2 0 1 5 7 3 6 8 >
< 7 0 1 4 5 3 2 6 8 | 21 | 0 7 1 4 5 3 2 6 8 >
< 1 4 2 3 5 0 7 6 8 | 23 | 1 4 2 3 5 7 0 6 8 >
< 4 2 7 1 5 0 3 6 8 | 25 | 4 2 0 1 5 7 3 6 8 >
~~~~~~~~~~~~~~
This is Close List:
< 0 0 0 0 0 0 0 0 0 | 0 | NULL >
< 1 4 0 2 5 7 3 6 8 | 14 | 1 4 7 2 5 0 3 6 8 >
< 4 7 0 1 5 3 2 6 8 | 14 | 4 0 7 1 5 3 2 6 8 >
< 0 4 2 1 5 7 3 6 8 | 16 | 1 4 2 0 5 7 3 6 8 >
< 0 4 7 1 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 4 0 7 5 3 2 6 8 | 16 | 1 4 7 0 5 3 2 6 8 >
< 1 0 4 2 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 1 4 2 0 5 7 3 6 8 | 17 | 1 4 0 2 5 7 3 6 8 >
< 4 0 2 1 5 7 3 6 8 | 17 | 0 4 2 1 5 7 3 6 8 >
< 1 4 7 0 5 3 2 6 8 | 17 | 1 4 7 2 5 3 0 6 8 >
```

```
< 4 0 7 1 5 3 2 6 8 | 17 | 0 4 7 1 5 3 2 6 8 >
< 4 7 3 1 5 0 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 4 7 1 0 5 3 2 6 8 | 17 | 4 7 0 1 5 3 2 6 8 >
< 1 4 7 2 5 3 0 6 8 | 18 | 1 4 7 2 5 0 3 6 8 >
< 0 1 4 2 5 7 3 6 8 | 18 | 1 0 4 2 5 7 3 6 8 >
< 1 4 2 3 5 7 0 6 8 | 18 | 1 4 2 0 5 7 3 6 8 >
< 4 2 0 1 5 7 3 6 8 | 18 | 4 0 2 1 5 7 3 6 8 >
< 4 7 3 1 0 5 2 6 8 | 18 | 4 7 3 1 5 0 2 6 8 >
< 0 7 1 4 5 3 2 6 8 | 18 | 4 7 1 0 5 3 2 6 8 >
< 1 4 7 2 5 0 3 6 8 | 19 | 1 4 7 2 5 8 3 6 0 >
< 1 4 7 2 5 8 3 6 0 | 20 | NULL >
~~~~~~~~~~~~~~
```

<u>Outfile 2 (results) second pair</u>

```
Solution found.
Goal
3 6 0
1 4 7
2 5 8

3 6 1
0 4 7
2 5 8

3 6 1
4 0 7
2 5 8

3 0 1
4 6 7
2 5 8

0 3 1
4 6 7
2 5 8

4 3 1
0 6 7
2 5 8

4 3 0
1 6 7
2 5 8
```

```
4 3 7
1 6 0
2 5 8

4 3 7
1 0 6
2 5 8

4 0 7
1 3 6
2 5 8

0 4 7
1 3 6
2 5 8

1 4 7
0 3 6
2 5 8

1 4 7
2 3 6
0 5 8

1 4 7
2 3 0
6 5 8

1 4 7
2 0 3
6 5 8

1 4 7
2 5 3
6 0 8

1 4 7
2 5 3
0 6 8

1 4 7
2 5 0
3 6 8
```

1 4 7
2 5 8
3 6 0

Start Node