**Chandigarh Group of Colleges**

**College of Engineering, Landran, Mohali-140307**

Department of Computer Science & Engineering



**ARTIFICIAL INTELLIGENCE**

**(**BTCS 605-18**)**

SUBMITTED TO SUBMITTED BY   
Ms.Reecha Sood Abhinandan  
 1914565

CSE

6th Semester

Group – New

INDEX

|  |  |  |
| --- | --- | --- |
| S.NO. | PRACTICAL | PAGE NO. |
| 1 | Introduction of Artificial Intelligence and its application. | 2-6 |
| 2 | Implementation of Depth-First Search(DFS). | 7-9 |
| 3 | Write a Program to implement Towers of Hanoi. | 10-11 |
| 4 | Implementation of Breadth- First Search (BFS). | 12-14 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Practical No. :- 01**

AIM: -Introduction of Artificial Intelligence and its application.

**Artificial intelligence:-** “The science and engineering of making intelligent machines, especially intelligent computer programs”.

Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think. AI is a study of how human brain think, learn, decide and work, when it tries to solve problems. And finally this study outputs intelligent software systems. The aim of AI is to improve computer functions which are related to human knowledge, for example, reasoning, learning, and problem-solving.

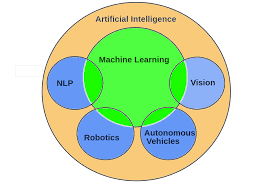
The intelligence is intangible. It is composed of

* Reasoning
* Learning
* Problem Solving
* Perception
* Linguistic Intelligence



The objectives of AI research are reasoning, knowledge representation, planning, learning, natural language processing, realization, and ability to move and manipulate objects. There are long-term goals in the general intelligence sector.

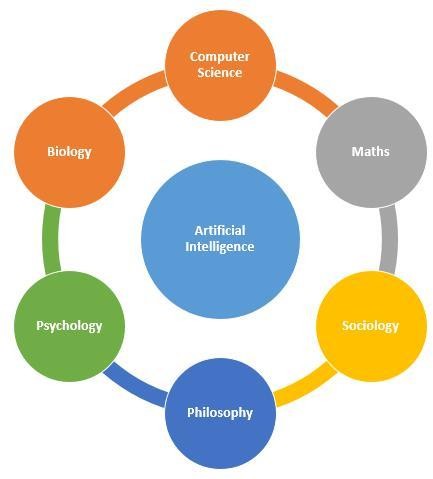
Approaches include statistical methods, computational intelligence, and traditional coding AI. During the AI research related to search and mathematical optimization, artificial neural networks and methods based on statistics, probability, and economics, we use many tools. Computer science attracts AI in the field of science, mathematics, psychology, linguistics, philosophy and so on.



**Major Goals:-**

* Knowledge reasoning
* Planning
* Machine Learning
* Natural Language Processing
* Computer Vision
* Robotics

## Foundation of AI:



**Applications of AI :-**

* Gaming − AI plays important role for machine to think of large number of possible positions based on deep knowledge in strategic games. for example, chess,river crossing, N-queens problems and etc.
* Natural Language Processing − Interact with the computer that understands natural language spoken by humans.
* Expert Systems − Machine or software provide explanation and advice to the users.
* Vision Systems − Systems understand, explain, and describe visual input on the computer.
* Speech Recognition − There are some AI based speech recognition systems have ability to hear and express as sentences and understand their meanings while a person talks to it. For example Siri and Google assistant.
* Handwriting Recognition − The handwriting recognition software reads the text written on paper and recognize the shapes of the letters and convert it into editable text.
* Intelligent Robots − Robots are able to perform the instructions given by a human.

**Advantages of Artificial Intelligence:-**

* AI would have a low error rate compared to humans, if coded properly. They would have incredible precision, accuracy, and speed.
* They won't be affected by hostile environments, thus able to complete dangerous tasks, explore in space, and endure problems that would injure or kill us.
* This can even mean mining and digging fuels that would otherwise be hostile for humans.
* Replace humans in repetitive, tedious tasks and in many laborious places of work.
* Predict what a user will type, ask, search, and do. They can easily act as assitants and cna recommend or direct various actions.
* An example of this can be found in the smartphone.
* Can detect fraud in card-based systems, and possibly other systems in the future.
* Organized and manages records.
* Interact with humans for entertainment or a task as avatars or robots.
* An example of this is AI for playing many videogames.
* Robotic pets can interact with humans. Can help w/ depression and inactivity.
* Can fulfill sexual pleasure.
* They can think logically without emotions, making rational decisions with less or no mistakes.
* Can assess people.
* This can be for medical purposes, such as health risks and emotional state. Can simulate medical procedures and give info on side effects.
* Robotic radiosurgery, and other types of surgery in the future, can achieve precision that humans can't.
* They don't need to sleep, rest, take breaks, or get entertained, as they don't get bored or tired.

**Disadvantages of Artificial Intelligence:-**

* Can cost a lot of money and time to build, rebuild, and repair. Robotic repair can occur to reduce time and humans needing to fix it, but that'll cost more money and resources.
* It's questionable: is it ethically and morally correct to have androids, human-like robots, or recreate intelligence, a gift of nature that shouldn't be recreated? This is a discussion about AI that's popular in the days.
* Storage is expansive, but access and retrieval may not lead to connections in memory as well as humans could.
* They can learn and get better with tasks if coded to, but it's questionable as to if this can ever become as good as humans can do such.
* They cannot work outside of what they were programmed for.
* They could never, or, at least, seemingly never with our technological perceptions, recieve creativity that humans have.
* This can prevent sympathizing with emotions for human contact, such as in being nurses. This can also reduce wisdom can understanding.
* This can prevent common sense occuring. Even if coded with common sense and to learn, it seems hard for them to get as much common sense that humans could.
* Robots, with them replacing jobs, can lead to severe unemployment, unless if humans can fix the unemployment with jobs AI can't do or severly change the government to communism.
* As seen partially with smartphones and other technology already, humans can become too dependent on AI and lose their mental capacities.
* Machines can easily lead to destruction, if put in the wrong hands. That is, at least a fear of many humans.
* AI as robots can supercede humans, enslaving us.

**Practical No. :- 02**

**AIM: -Implementation of Depth-First Search(DFS).**

**Depth-First Search**

DFS is also an important type of uniform search. DFS visits all the vertices in the graph. This type of algorithm always chooses to go deeper into the graph. After DFS visited all the reachable vertices from a particular sources vertices it chooses one of the remaining undiscovered vertices and continues the search. DFS reminds the space limitation of breath first search by always generating next a child of the deepest unexpanded nodded. The data structure stack or (LIFO) is used for DFS. One interesting property of DFS is that, the discover and finish time of each vertex from a parenthesis structure. If we use one open parenthesis when a vertex is finished then the result is properly nested set of parenthesis.

## Advantages of Depth-First Search

* DFS consumes very less memory space.
* It will reach at the goal node in a less time period than BFS if it traverses in a right path.
* It may find a solution without examining much of search because we may get the desired solution in the very first go.
* It takes less memory as compared to BFS as BFS requires entire tree to be stored but this requires only one path.
* Sometimes solution lies in earlier stages then DFS is better.
* If there are multiple solutions then DFS stops when first solution is found. Where as BFS gives all the solutions at the same time.

## Disadvantages of Depth-First Search

* There is a possibility that it may go down the left-most path forever. Even a finite graph can generate an infinite tree.
* It is possible that may states keep reoccurring. There is no guarantee of finding the goal node.
* Sometimes the states may also enter into infinite loops.
* Depth-First Search is not guaranteed to find the solution.
* And there is no guarantee to find a minimal solution, if more than one solution exists.

## Algorithm of Depth-First Search

1. PUSH the starting node into the stack.
2. If the stack is empty then stop and return failure.
3. If the top node of the stack is the goal node, then stop and return success.
4. Else POP the top node from the stack and process it. Find all its neighbors that are in ready state and PUSH them into the stack in any order.
5. Go to step 3.
6. Exit.

## Program: -

#include<iostream>

using namespace std;

int cost[10][10],i,j,k,n,stk[10],top,v,visit[10],visited[10];

int main()

{

int m;

cout<<"Enter of vertices:";

cin>>n;

cout<<"Enter of edges:";

cin>>m; cout<<"EDGES\n";

for(k=1;k<=m;k++)

{

cin>>i>>j;

cost[i][j]=1;

}

cout<<"Enter initial vertex:";

cin>>v;

cout<<"Order of Visited vertices:\n";

cout<<v<<" ";

visited[v]=1; k=1;

while(k<n)

{

for(j=n;j>=1;j--)

if(cost[v][j]!=0&& visited[j]!=1&& visit[j]!=1)

{

visit[j]=1;

stk[top]=j;

top++;

}

v=stk[--top];

cout<<v<<" "; k++;

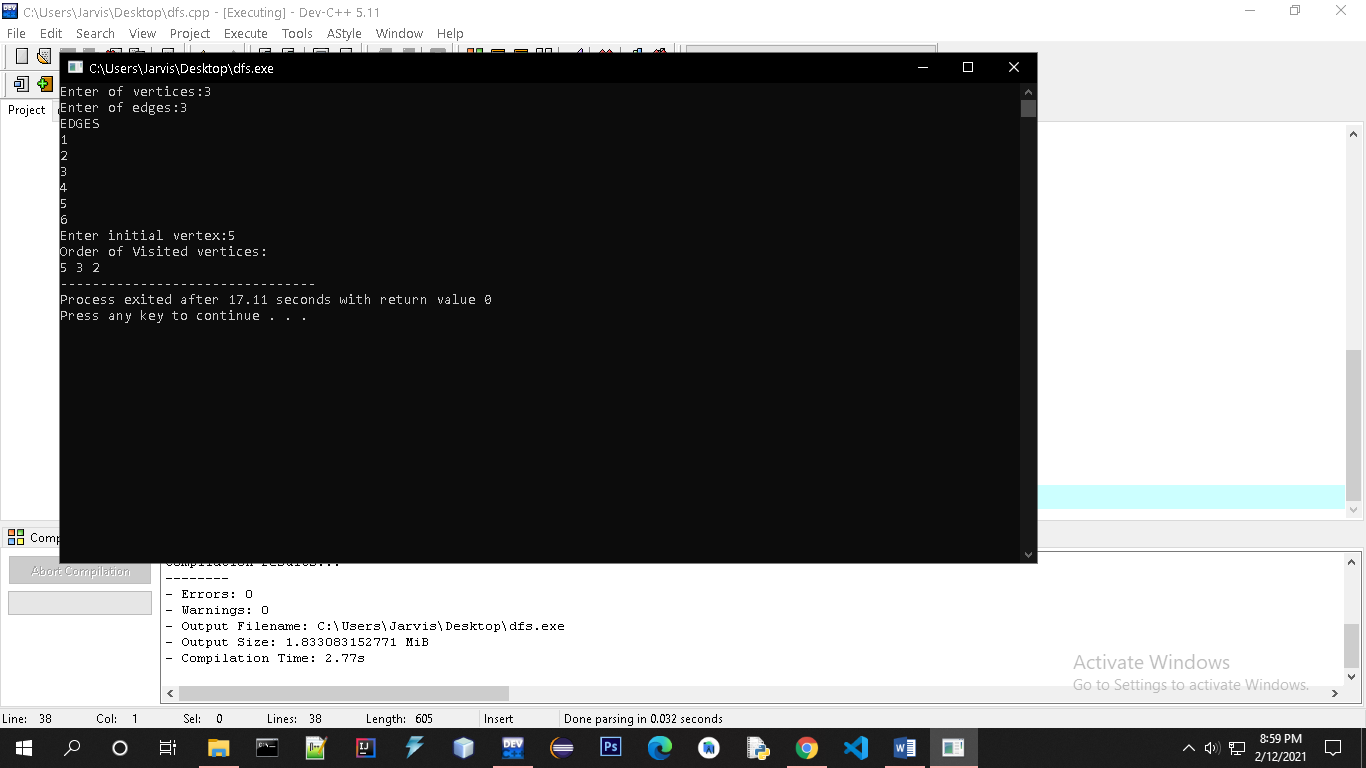
visit[v]=0;

visited[v]=1;

}

return 0;

}



**PRACTICAL NO: 3**

AIM:- Write a Program to implement Towers of Hanoi.

Tower of Hanoi

The tower of Hanoi is a mathematical puzzle. It consists of three rods and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top. We have to obtain the same stack on the third rod.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules−

* Only one disk can be moved at a time.
* Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
* No disk may be placed on top of a smaller disk.

## Algorithm of Tower of Hanoi

## If we have only one disk , then it can easily be moved from source to destination peg.

## If we have 2 disks −

* First, we move the smaller (top) disk to aux peg.
* Then, we move the larger (bottom) disk to destination peg.
* And finally, we move the smaller disk from aux to destination peg.
* **To move n disks from source to destination, The steps to follow are -**
* Step 1 − Move n-1 disks from source to aux
* Step 2 − Move nth disk from source to dest
* Step 3 − Move n-1 disks from aux to dest

**Program:**

#include <stdio.h>

#include <conio.h>

void toh(int, char, char, char);

int main()

{ int n;

printf(" Abhinandan 1914565");

printf("\n Enter the number of disk : ");

scanf("%d", &n);

printf("Here is sequence of moves of tower of hanoi :\n");

toh(n, 'A', 'C', 'B');

}

void toh(int no, char source, char destination, char spare)

{

if (no == 1) {

printf("\n move disk 1 from source %c to destination %c", source, destination);

return; }

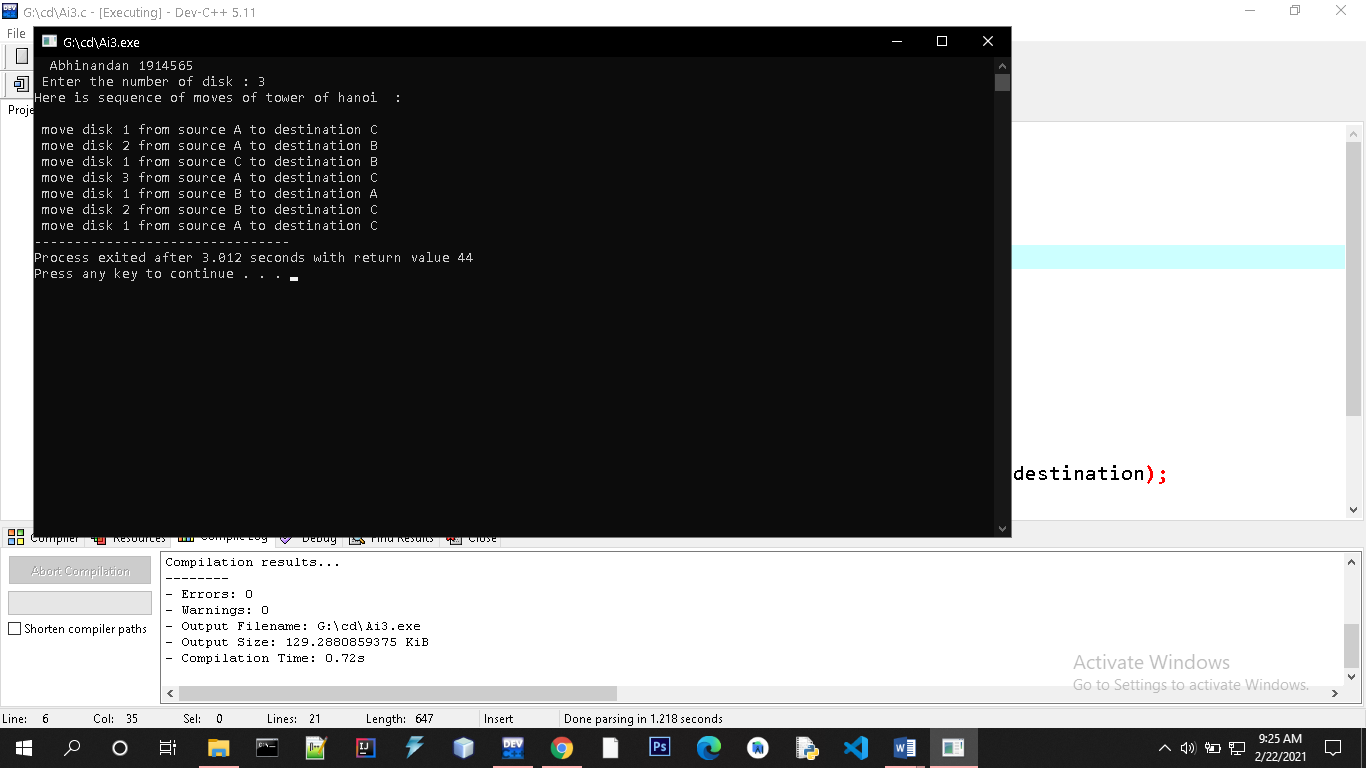
toh(no - 1, source, spare, destination);

printf("\n move disk %d from source %c to destination %c", no, source, destination);

toh(no - 1, spare, destination, source);

}

**Output:-**

****

**Practical No. :- 04**

AIM: -Implementation of Breadth-First Search(BFS).

Breadth-First Search:-

Breadth-first search (BFS) is an algorithm that is used to graph data or searching tree or traversing structures. The full form of BFS is the Breadth-first search.

The algorithm efficiently visits and marks all the key nodes in a graph in an accurate breadthwise fashion. This algorithm selects a single node (initial or source point) in a graph and then visits all the nodes adjacent to the selected node. Remember, BFS accesses these nodes one by one.

Once the algorithm visits and marks the starting node, then it moves towards the nearest unvisited nodes and analyses them. Once visited, all nodes are marked. These iterations continue until all the nodes of the graph have been successfully visited and marked.

## Advantages of Depth-First Search

* Used to find the shortest path between vertices
* Always finds optimal solutions.
* There is nothing like useless path in BFS,since it searches level by level.
* Finds the closest goal in less time

## Disadvantages of Depth-First Search

* Memory Constraints As it stores all the nodes of the present level to go for the next level.
* If a solution is far away then it consumes time.

Algorithm of Breadth-First Search:-

* **Step 1:** SET STATUS = 1 (ready state) for each node in G
* **Step 2:** Enqueue the starting node A and set its STATUS = 2  
  (waiting state)
* **Step 3:** Repeat Steps 4 and 5 until QUEUE is empty
* **Step 4:** Dequeue a node N. Process it and set its STATUS = 3  
  (processed state).
* **Step 5:** Enqueue all the neighbours of N that are in the ready state  
  (whose STATUS = 1) and set their STATUS = 2 (waiting state)  
  [END OF LOOP]
* **Step 6:** EXIT

Program:-

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

int cost[10][10],i,j,k,n,qu[10],front,rare,v,visit[10],visited[10];

int main()

{

int m;

cout<<"Enter no of vertices:";

cin>>n;

cout<<"Enter no of edges:";

cin>>m;

cout<<"\nEDGES \n";

for(k=1; k<=m; k++)

{

cin>>i>>j;

cost[i][j]=1;

}

cout<<"Enter initial vertex to traverse from:";

cin>>v;

cout<<"Visitied vertices:";

cout<<v<<" ";

visited[v]=1;

k=1;

while(k<n)

{

for(j=1; j<=n; j++)

if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)

{

visit[j]=1;

qu[rare++]=j;

}

v=qu[front++];

cout<<v <<" ";

k++;

visit[v]=0;

visited[v]=1;

}

return 0;

}

Output:-

