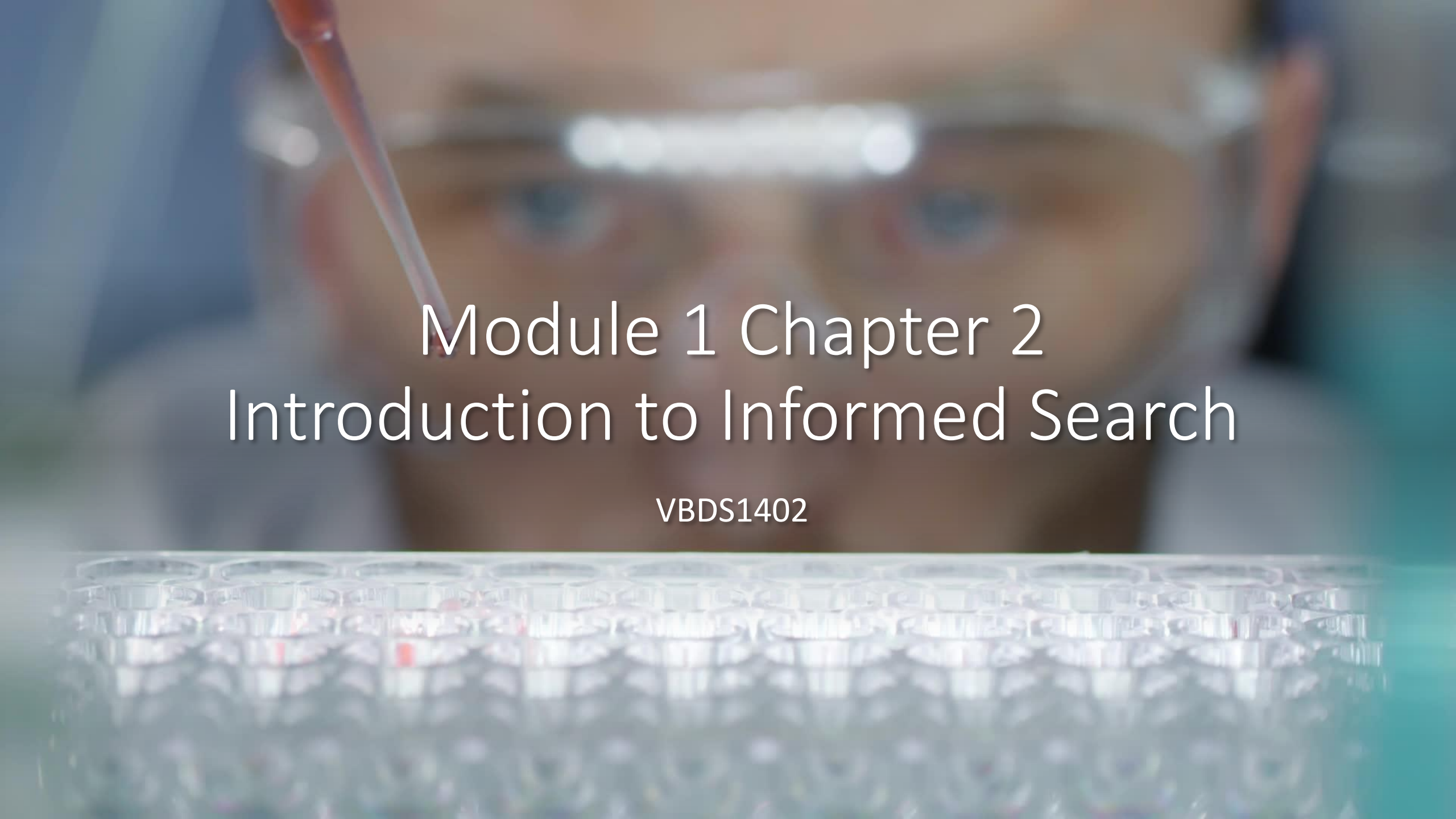


Module 1 Chapter 2
Introduction to Informed Search

VBDS1402

In this session you will learn:

- Basics of Informed search
- Types of informed search
- Greedy best first search
- Greedy best first search example

Informed Search

Contains an array of knowledge such as how far we are from the goal, path cost, how to reach to goal node, etc.

It helps to explore less to the search space and find more efficiently the goal node.

Useful for large search space.

Uses the idea of heuristic, so it is also called Heuristic search.

General approach of Informed Search (Best First Search)

Node selected for expansion based on evaluation function: f(n)

The evaluation function is a cost estimate, so the node with the lowest evaluation is expanded first.

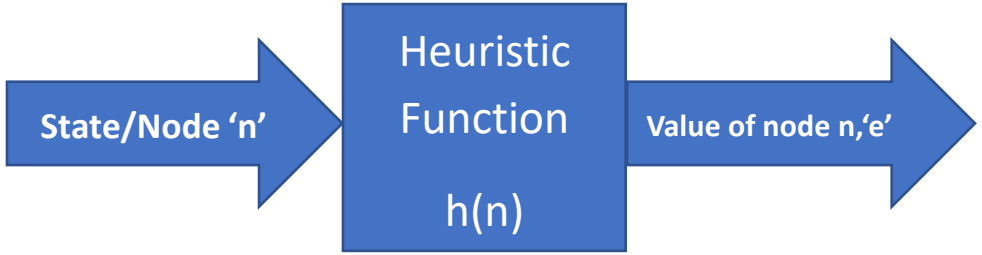
Implementation is identical to that for uniform-cost search.

Choice of f determines the search strategy.

h (n): HEURISTIC FUNCTION h (n)= estimated cost of the cheapest path from the state at node n to a goal state.

Heuristic function

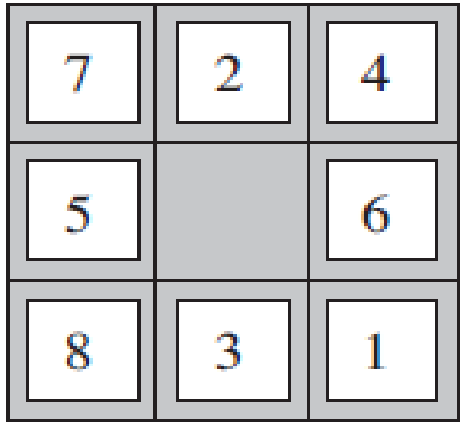
- An evaluation function, to which the search state is given as input and it generates the tangible representation of the state as output.
- Evaluates individual problem state and determines how much promising the state is.
- h(n)=estimated cost of the cheapest path from the state at node n to a goal state.
- Example: For the Travelling Salesman Problem, the sum of the distances travelled so far can be a simple heuristic function.



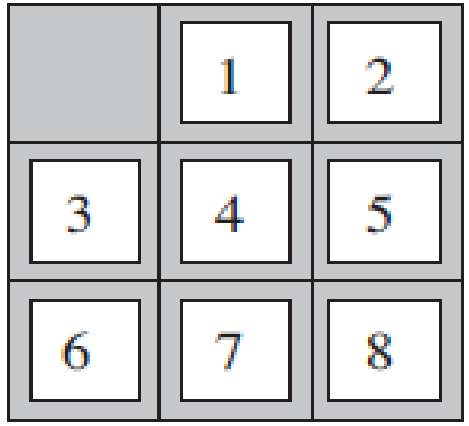
- Might not always give the best solution, but it guaranteed to find a good solution in reasonable time.
- Value of the heuristic function is always positive.
- h(n) <= h*(n)
  - where h(n) is heuristic cost, and h*(n) is the estimated cost.

Heuristic function Example

- h1 = the number of misplaced tiles.
  - h1 = 8.
- h2 = the sum of the distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of the horizontal and vertical distances.
  - Manhattan distance of h2 = 3+1 + 2 + 2+ 2 + 3+ 3 + 2 = 18.



Start State



Goal State

Types of Informed Search

GREEDY BEST-FIRST SEARCH

A* SEARCH

MEMORY-BOUNDED HEURISTIC SEARCH

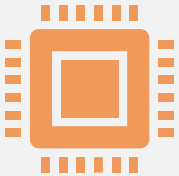
Types of Informed Search

Greedy best-first search tries to expand the node that is closest to the goal, on the ground that this is likely to lead to a solution quickly.

Heuristic function: f(n)=h(n).

A* evaluates nodes by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal: f(n)=g(n)+h(n) .

Identical to UNIFORM-COST-SEARCH except that A∗ uses g + h instead of g.



The simplest way to reduce memory requirements for A∗ is to use iterative-deepening A∗ (IDA∗) algorithm.

Difference between IDA∗ and IDDFS: is that the cut off used is the f-cost (g +h) rather than the depth

Best first search algorithm (Greedy)

Step 1	Place the starting node into the OPEN list.
Step 2	If the OPEN list is empty, Stop and return failure.
Step 3	Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.
Step 4	Expand the node n, and generate the successors of node n.
Step 5	Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.
Step 6	For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.
Step 7	Return to Step 2.

Best First Search

Straight Line distance (Euclidean) Heuristic

A->G=40

B->G=32

C->G=25

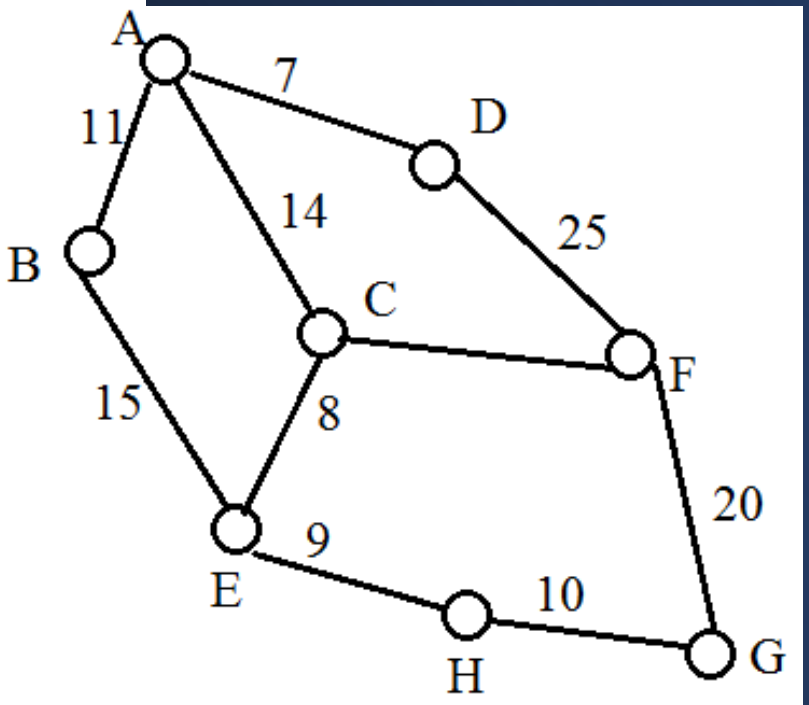
D->G=35

E->G=19

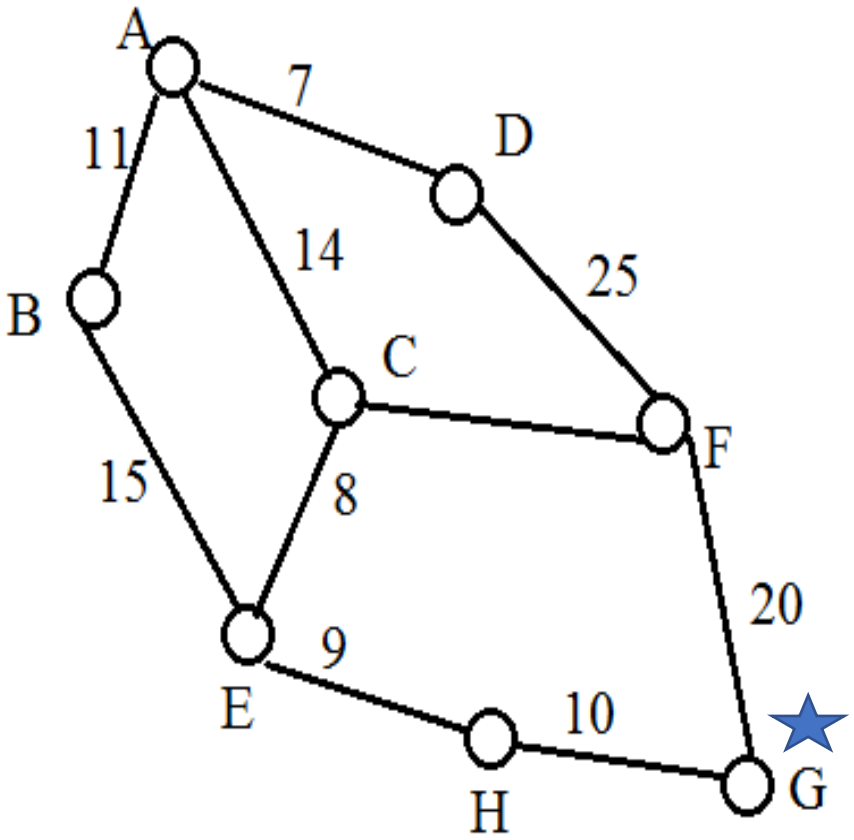
F->G=17

H->G=10

G->G=0



Best First Search



Heuristic:

A->G=40

B->G=32

C->G=25

D->G=35

E->G=19

F->G=17

H->G=10

G->G=0

- open=[A], closed=[]
- open=[C25, B32,D35], closed=[A]
- open=[F17,E19,B32,D35],closed=[A,C]
- open=[G0,E19, B32,D35], closed=[A,C,F]

Goal node reached

Path= A->C->F->G

Best First Search (Advantages)

- It is more efficient than that of BFS and DFS.
- Time complexity of Best first search is much less than Breadth first search.
- The Best first search allows us to switch between paths by gaining the benefits of both breadth first and depth first search.

Best First Search (Disadvantages)

It can behave as an unguided depth-first search in the worst -case scenario.

It can get stuck in a loop as DFS.

This algorithm is not optimal.

Difference between informed and Uninformed Search

Informed Search	Uninformed Search
It uses knowledge for the searching process.	It doesn’t use knowledge for searching process.
It finds solution more quickly.	It finds solution slow as compared to informed search.
It is highly efficient.	It is mandatory efficient.
Cost is low.	Cost is high.
It consumes less time.	It consumes moderate time.
It provides the direction regarding the solution.	No suggestion is given regarding the solution in it.
It is less lengthy while implementation.	It is more lengthy while implementation.
Greedy Search, A* Search, Graph Search	Depth First Search, Breadth First Search

Take home task

- Apply Best First Search