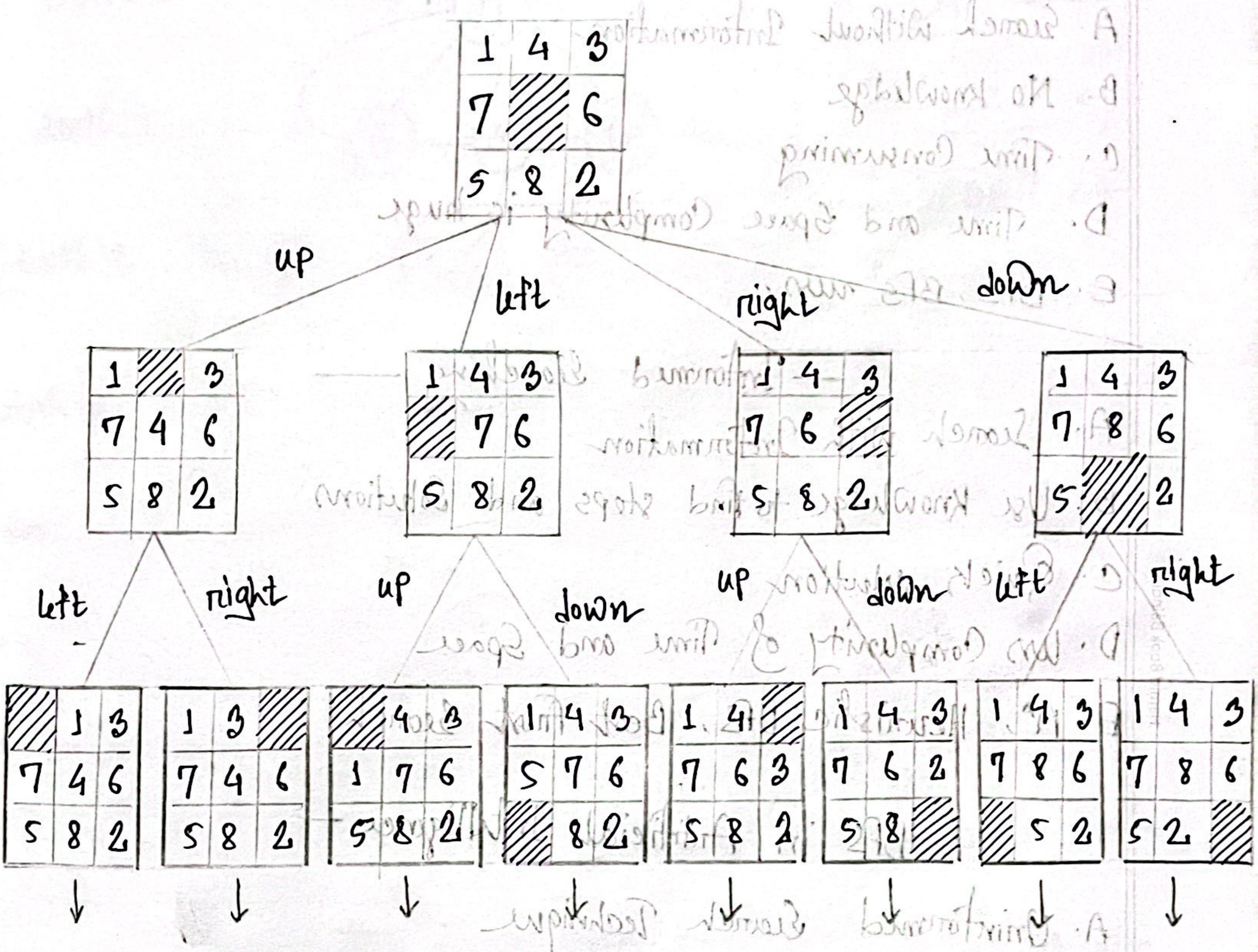
Subject :　　　　　　　　　　Date :

State Space Search



A*

Graph search → optimal → consistency
complete → admissible, consistency

Tree search → optimal
complete } → Admissible

Subject: Date:

— Uninformed Searching —

A. Search without Information

B. No knowledge

C. Time Consuming

D. Time and Space Complexity is huge

E. DFS, BFS uses.

— Informed Searching —

A. Search with Information

B. Use Knowledge to find steps and Solutions

C. Quick solution

D. less Complexity of Time and Space

E. A*, Heuristic DFS, Best First Search

— BFS in Artificial Intelligence —

A. Uninformed Search Technique

B. FIFO (Queue)

C Shallowest Node

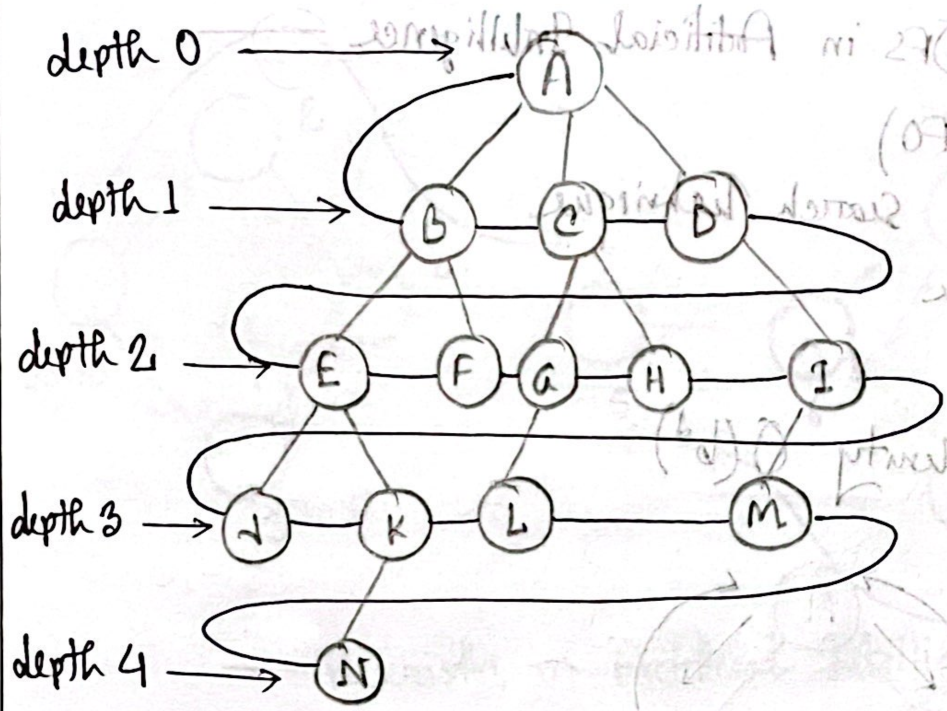
D. Complete

E Time Complexity $O(b^d)$

b = Branch Factor (Child number)

d = depth

Subject: Date:



A ←	~~A~~ ←
B ←	~~B~~CD ←
C ←	~~C~~DEF ←
D ←	~~D~~EFGH ←
E ←	~~E~~FGHI ←
F ←	~~F~~GHIJK ←
G ←	~~G~~HIJK ←
H ←	~~H~~IJKL ←
I ←	~~I~~JKL ←
J ←	~~J~~KLM ←
K ←	~~K~~LM ←
L ←	~~L~~MN ←
M ←	~~M~~N ←
N ←	~~N~~ ←

Queue

for child node = 3

Complexity $= 3^4$

$= 81$

depth 0 → 3 A Nodes

depth 1 – (3×3) for BCD Nodes

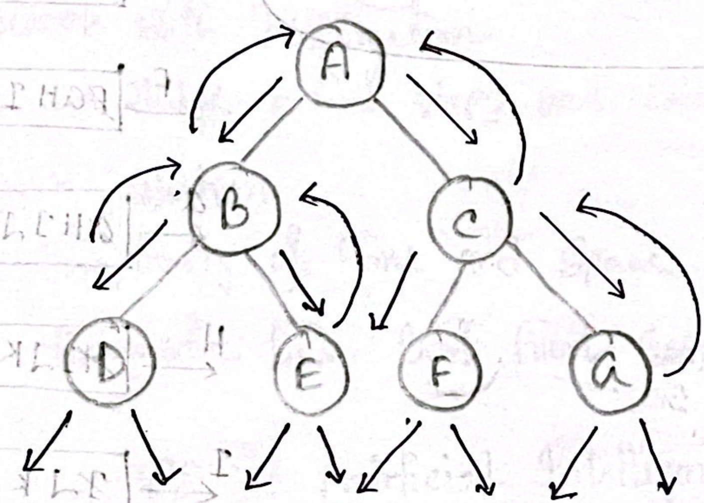
depth 2 – (3×9) for ~~..~~ 9 Nodes

depth 3 – (3×15) for 15 Nodes

Subject: Date:

DFS in Artificial Intelligence

A. Stack (LIFO)
B. Uninformed Search Technique
C. Deepest Node
D. Incomplete
E. Time Complexity $O(b^d)$



[A C G F B E D] Traversal Nodes Sequences

Bidirectional Search Algorithm

Two simultaneous search from on initial node to goal and backward from goal to initial, stopping when two meet.

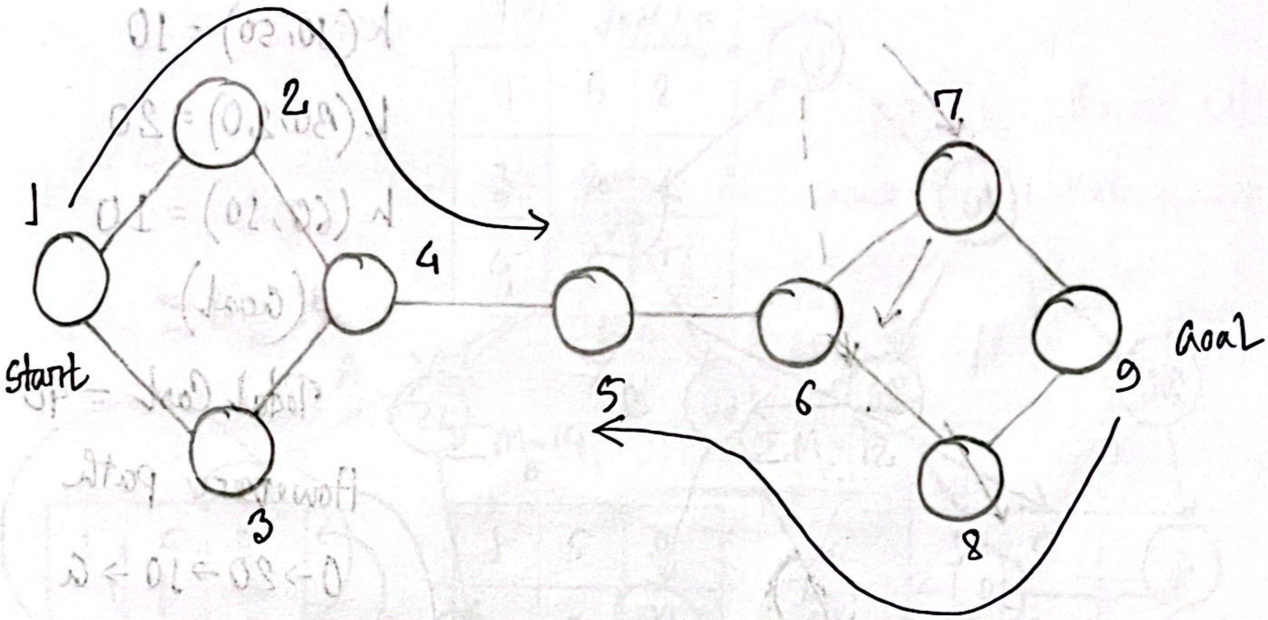
Time Complexity: $2(b^{d/2})$

Complete in breadth first search

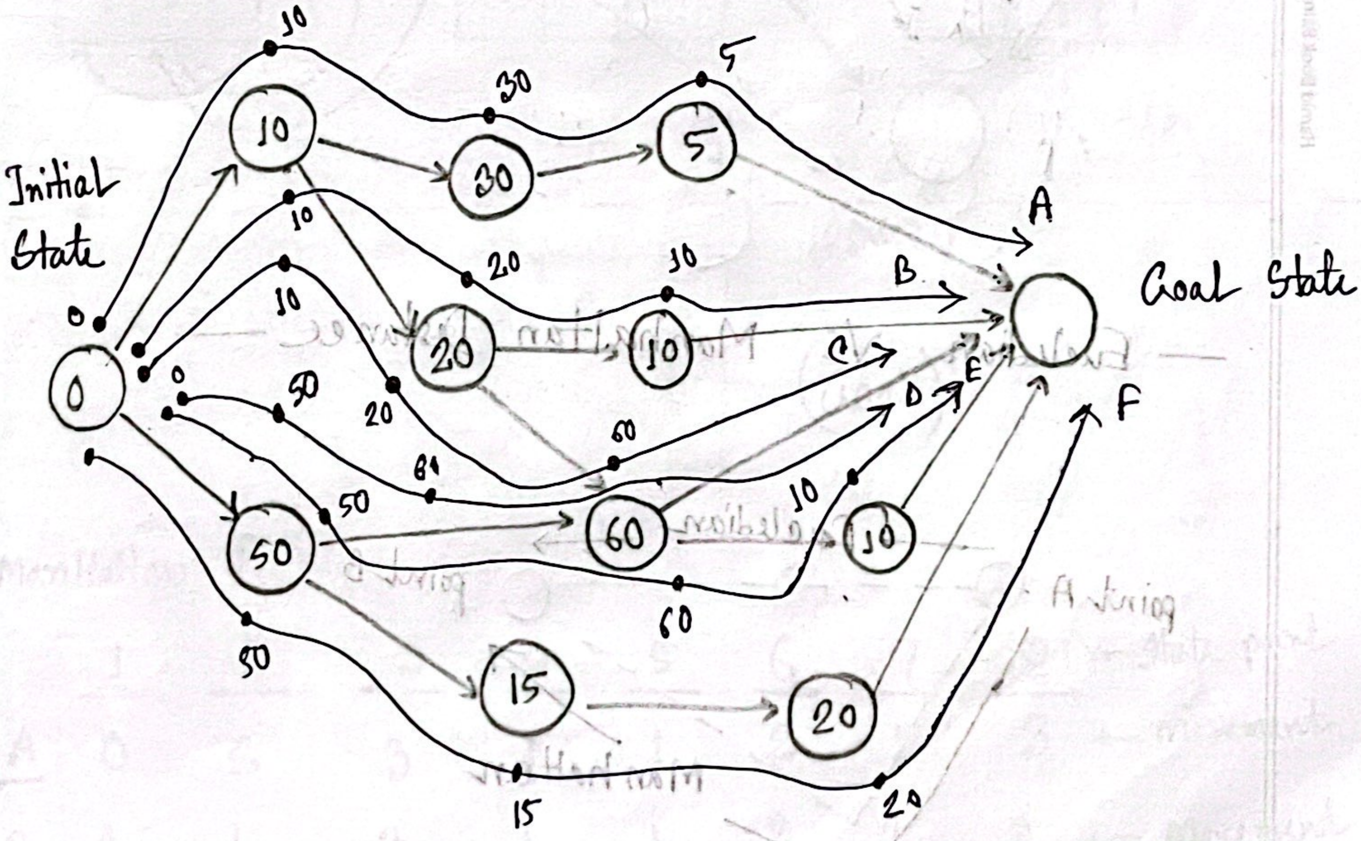
Not in depth first search

Subject :

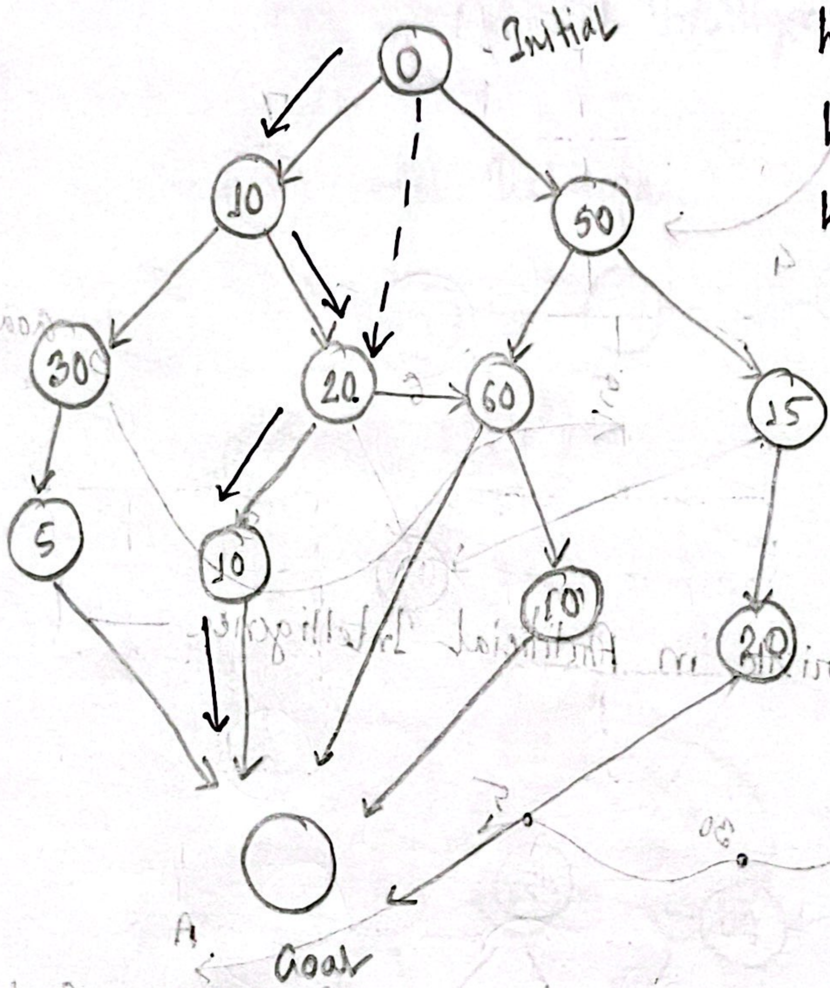
Date :



— Heuristic in Artificial Intelligence —



Subject : | Date :



$h(10,50) = 10$

$h(30,20) = 20$

$h(60,10) = 10$

(Goal)

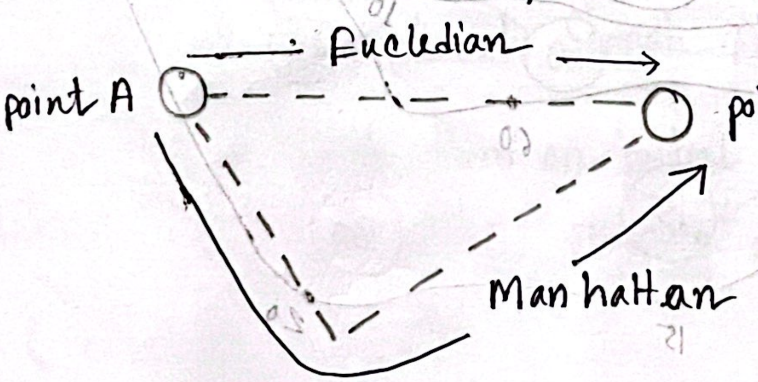
Total Cost = 40

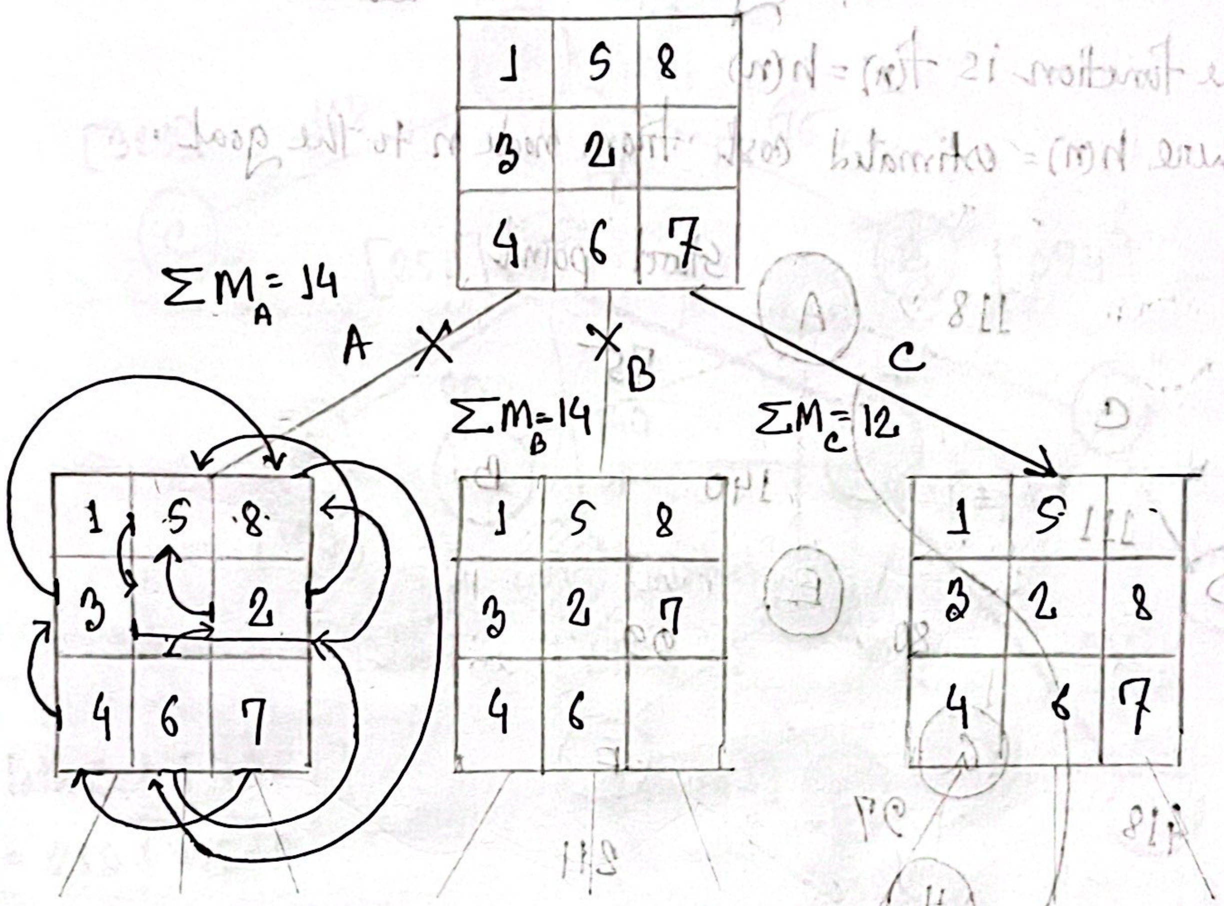
However path

$0 \to 20 \to 10 \to G$

Cost $30 < 40$

— Euclidian Vs Manhattan Distance —





1	2	3
4	5	6
7	8	

(Goal)

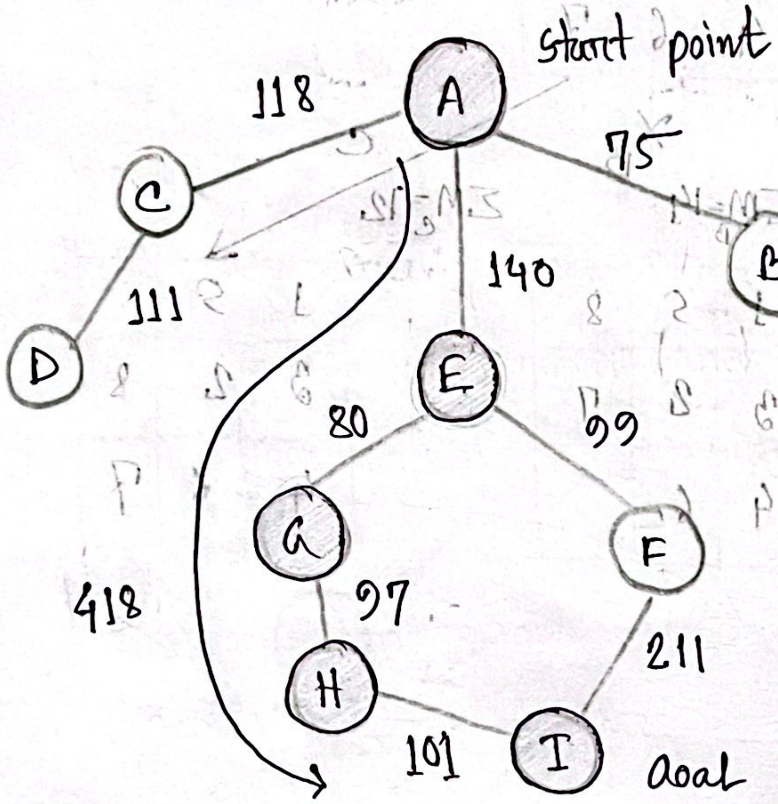
Manhattan Priority Function

	1	2	3	4	5	6	7	8	← state point
A	0	2	3	1	1	2	2	3	← Movement
B	0	1	3	1	1	2	3	3	← Movement
C	0	1	3	1	1	2	2	2	← Movement

— Greedy Best-First Search —

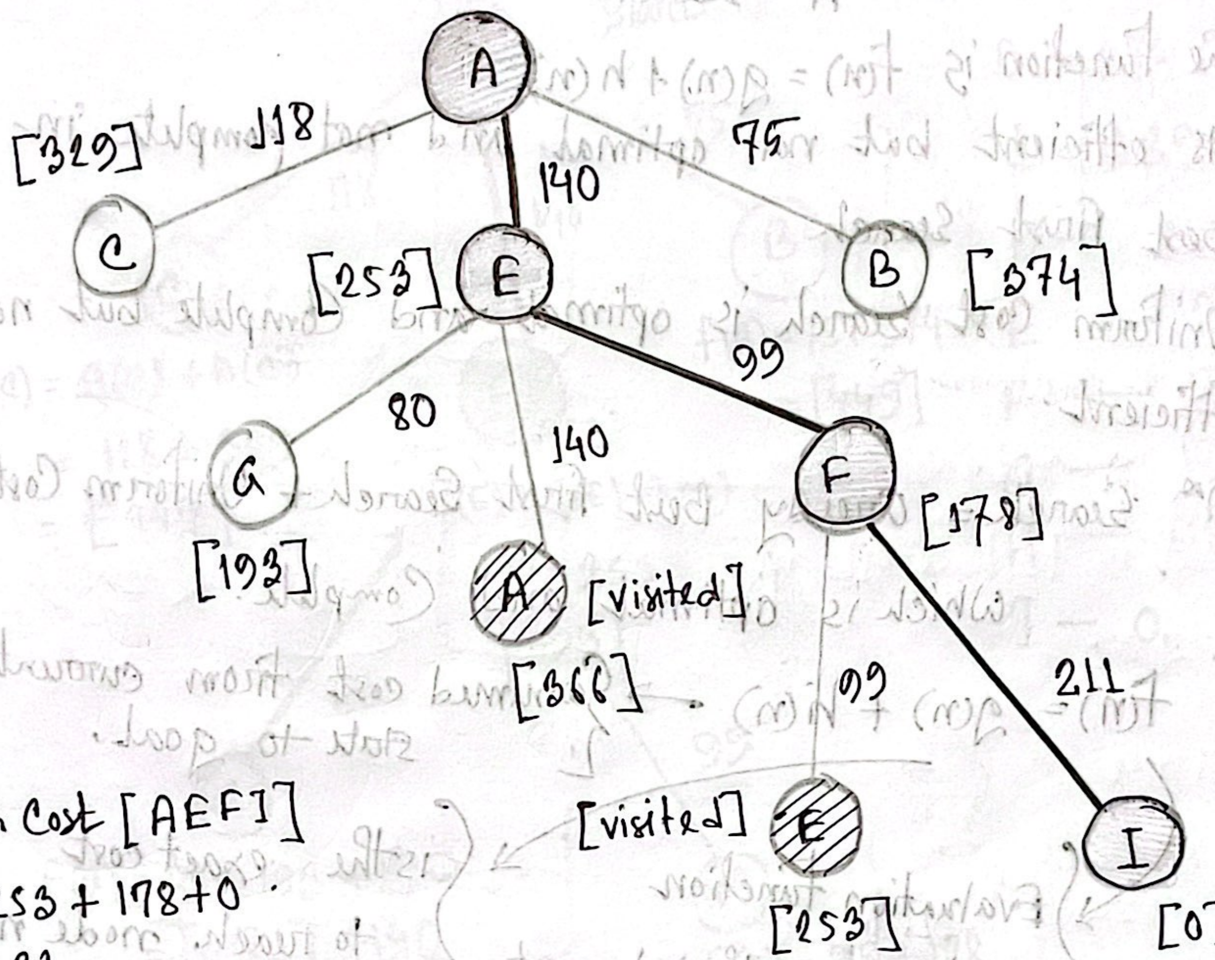
The function is $f(n) = h(n)$

Where $h(n)$ = estimated cost from node n to the goal.



State	Heuristic $h(n)$	State	Heuristic $h(n)$
A →	366	F →	178
B →	374	G →	193
C →	329	H →	98
D →	244	I →	0
E →	253		

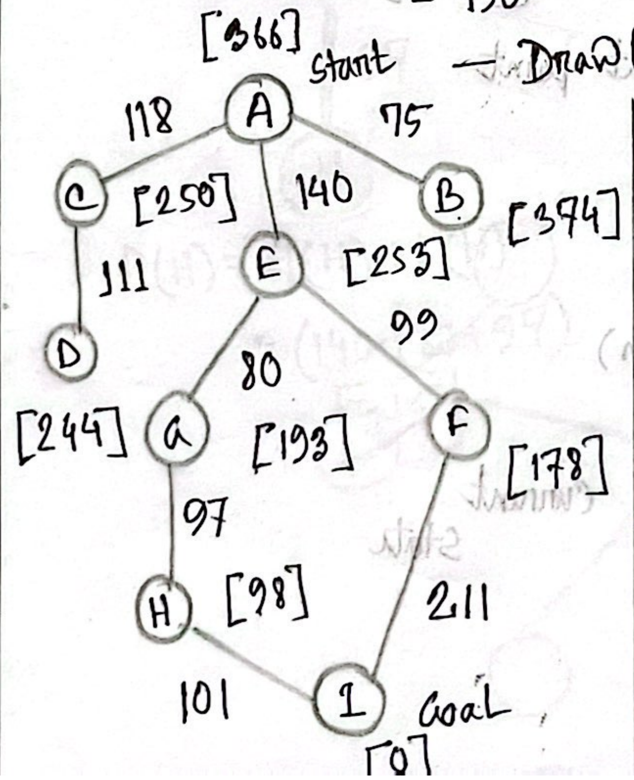
Subject : Date :

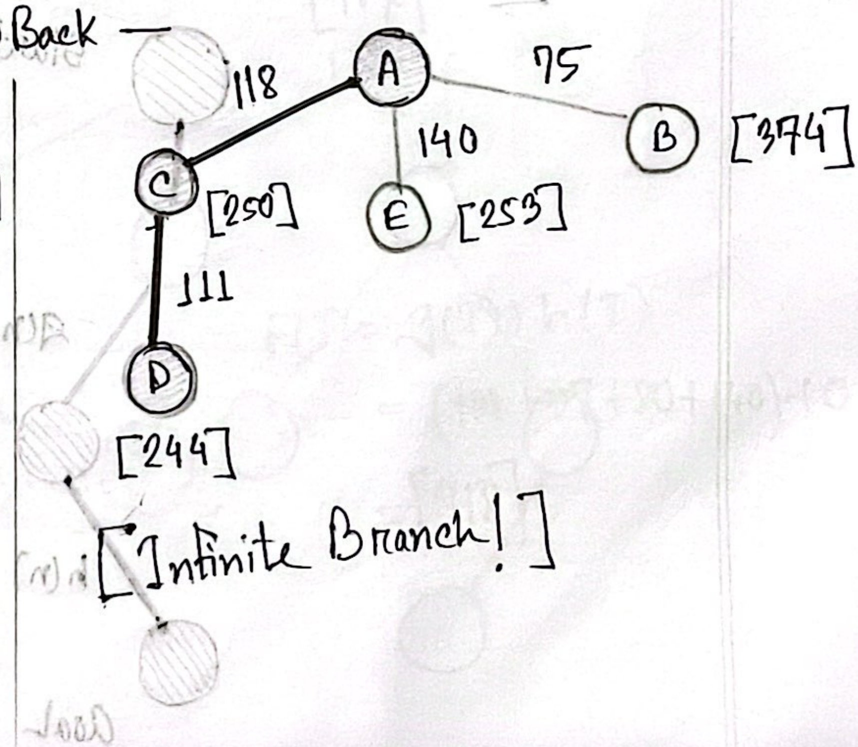


Path Cost [AEFI]
= 253 + 178 + 0
= 431

distance (AEFI) = 140 + 99 + 211
= 450

— Draw Back —





Subject: Date:

— A* Search —

The function is $f(n) = g(n) + h(n)$

~~It's~~ efficient but not optimal and not complete in Best First Search

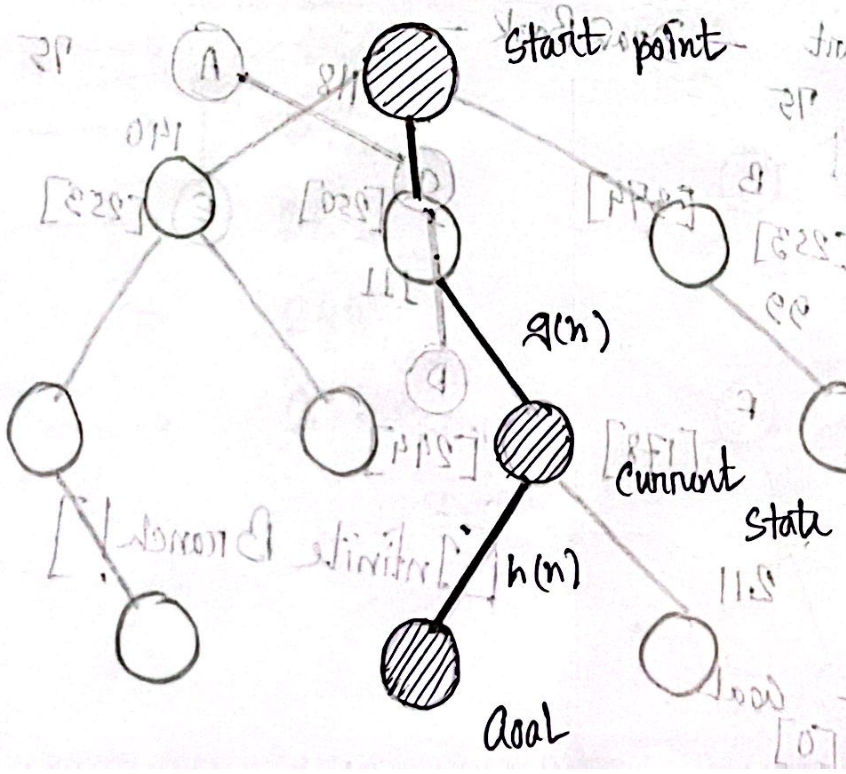
Uniform Cost Search is optimal and Complete but not efficient.

A* Search = Greedy Best first Search + Uniform Cost
which is optimal and Complete.

$f(n) = g(n) + h(n)$ → { assumed cost from current state to goal.

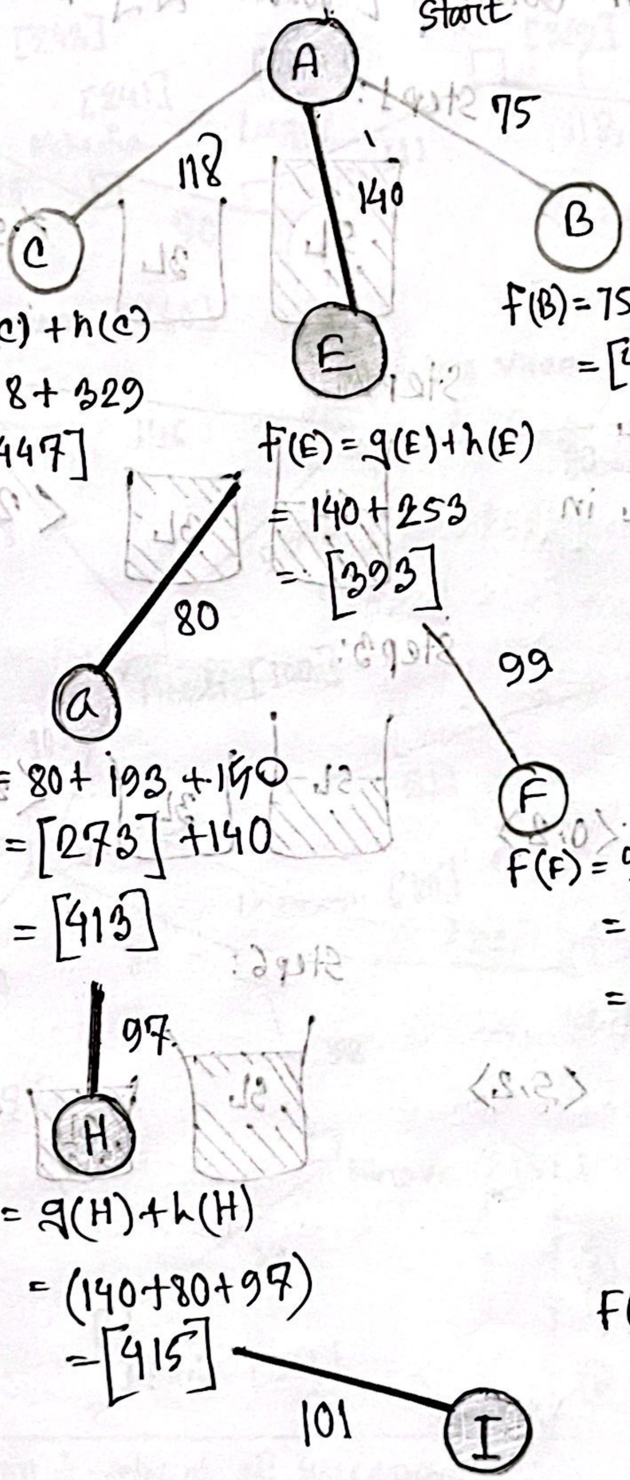
$f(n)$ → { Evaluation function which gives the cheapest solution cost

$g(n)$ → { is the exact cost to reach node n from the initial state.



Subject : | Date :

A — 366
B — 374
C — 329
D — 244
E — 253
F — 178
G — 193
H — 98
I — 0



$f(C) = g(C) + h(C)$
$= 118 + 329$
$= [447]$

$f(B) = 75 + 374$
$= [449]$

$f(E) = g(E) + h(E)$
$= 140 + 253$
$= [393]$

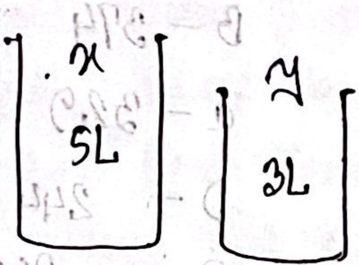
$f(G) = 80 + 193 + 140$
$= [273] + 140$
$= [413]$

$f(F) = 99 + 178$
$= [277] + 140$
$= [417]$

$f(H) = g(H) + h(H)$
$= (140 + 80 + 97)$
$= [415]$

$F(I) = g(I) + h(I)$
$= (101 + 97 + 80 + 140) + 0$
$= [418]$

Problem Solving [Water - Jug Problem]

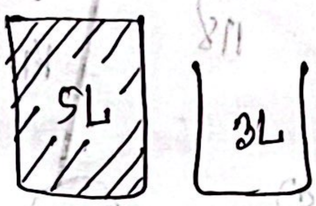


Jug x contains 5L

Jug y contains 3L

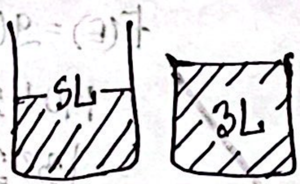
Goal is to have 4L in Jug x.

Step 1 :



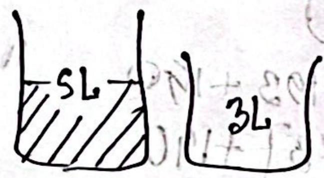
⟨5, 0⟩

Step 2 :



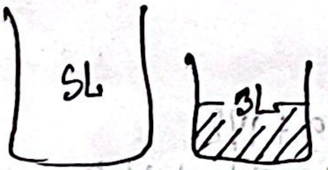
⟨2, 3⟩

Step 3 :



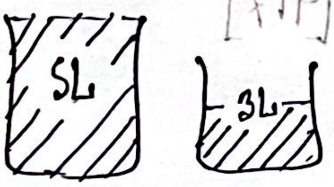
⟨2, 0⟩

Step 4 :



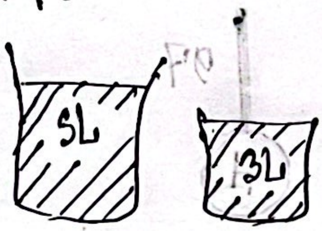
⟨0, 2⟩

Step 5 :



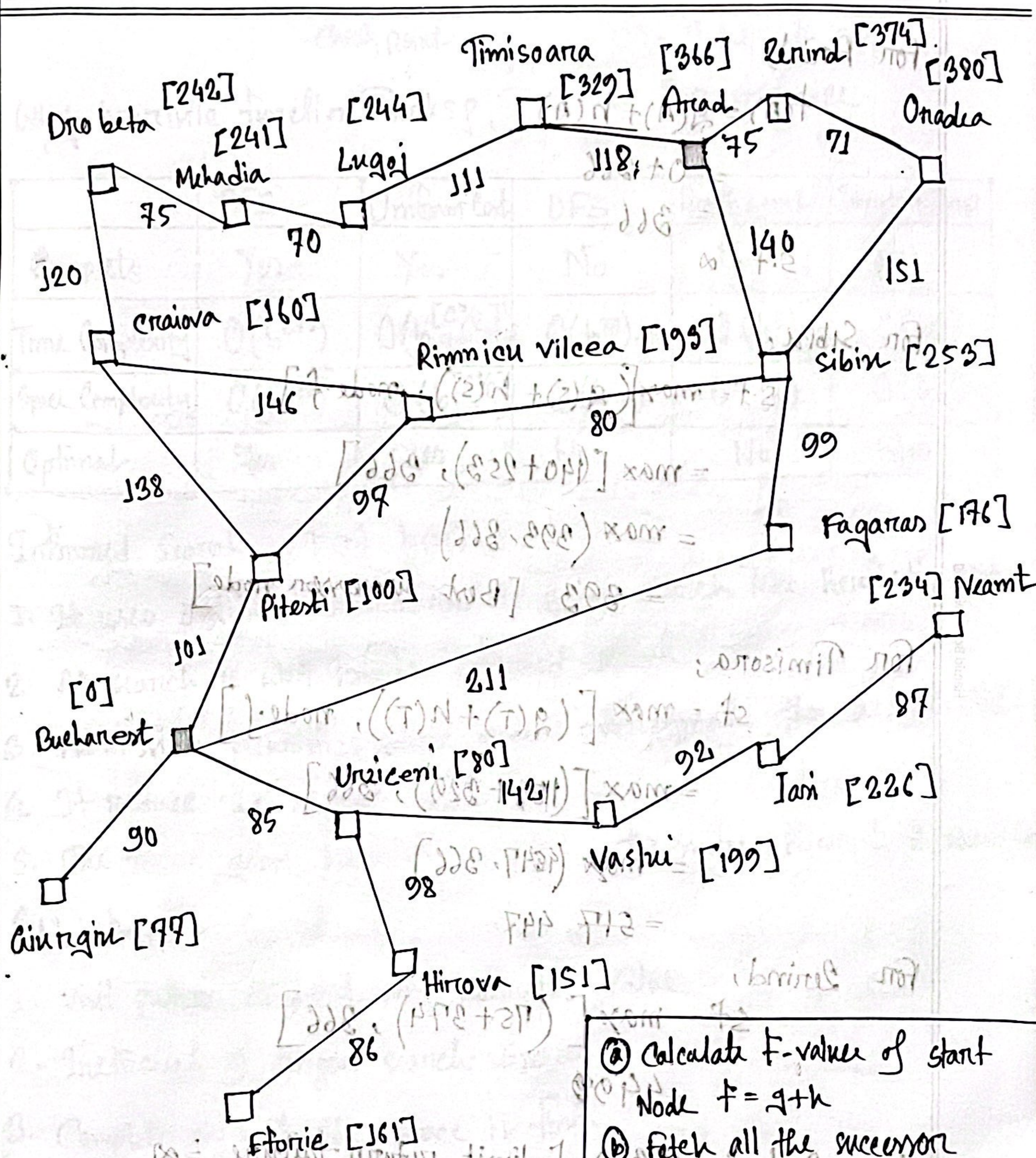
⟨5, 2⟩

Step 6 :



⟨4, 3⟩

Subject: # Romania Map AI Date:



ⓐ Calculate F-value of start Node $f = g+h$

ⓑ Fetch all the successor and calculate f value $f = g+h$

ⓒ e find max(f_s, f_{parent}) This is the new f value of succ.

ⓓ Find the min f-value of all successors

ⓔ

Subject: Date:

For Arad;

$$F(A) = g(A) + h(A)$$
$$= 0 + 366$$
$$= 366$$
$$s.f = \infty$$

For Sibiu;

$$s.f = \max[(g(s)+h(s)), node.f]$$
$$= \max[(140+253), 366]$$
$$= \max(393, 366)$$
$$= 393 \quad \text{[Best successor node]}$$

For Timisora;

$$sf = \max[(g(T)+h(T)), node.f]$$
$$= \max[(118+329), 366]$$
$$= \max(447, 366)$$
$$= \cancel{517}\ 447$$

For Zerind,

$$sf = \max[(75+374), 366]$$
$$= 449$$

Here $Best.f = 393 > f\_limit$ return failure $= \infty$

But $393 \not> \infty$

Alt.node $= \cancel{449}\ 447$

$f\_limit = \min(f\_limit, alt.f) = \min(\infty, 447) = \cancel{449}\ 447$

Subject: | Date:

Check point

C^* = Total path cost

ϵ = Avg step size.

Why heuristic function works?

	BFS	Uniform Cost	DFS	Depth Limit	Iterative Deep
Complete	Yes	Yes	No	No	Yes
Time Complexity	$O(b^{d+1})$	$O(b^{[C^*/\epsilon]})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space Complexity	$O(b^{d+1})$	$O(b^{[C^*/\epsilon]})$	$O(bm)$	$O(bl)$	$O(bd)$
Optimal	Yes	Yes	No	No	Yes

Informed Search : $F(n) = h(n)$

1. It uses additional information to guide search like heuristics value
2. A* search is well known informed search
3. Admissible heuristics value never overestimate the actual cost
4. It reduce the search state.
5. The more good heuristics value, the more efficient it would be

Uninformed Search:

1. visit paths without any heuristics value
2. Inefficient of larger search spaces.
3. Complete only if the space is finite
4. Find shortest path in BFS if the graph is unweighted
5. Explore the entire search space with worst case
6. Both Tree and graph search

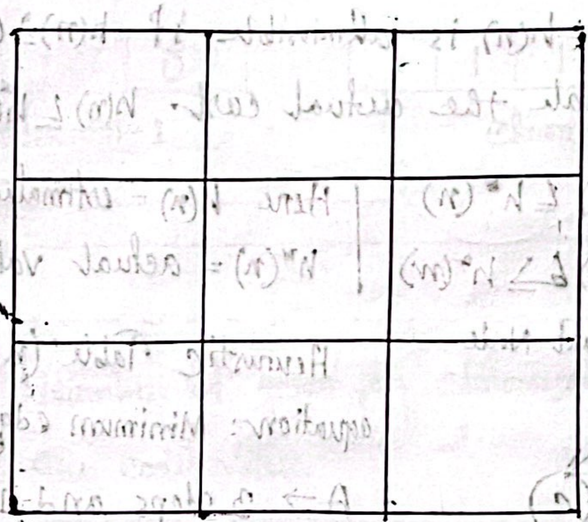
Subject: Date:

1. BFS is good for shortest path
2. DFS is good for deep search
3. A* for efficiency.
4. Due to heuristics value informed search uses less state than that of uninformed.
5. IDS (Iterative deepening search) uses DFS for low memory uses and BFS for completeness simultaneously.
6. A poor heuristics value can mislead the search and a good heuristics can ~~be~~ lead the ~~A*~~ search.
7. Overestimating the cost, may cause A* to skip the optimal path. So, heuristics never overestimate the actual path cost.
8. Bidirectional search ~~d~~ reduce the search space as it can search forword from the start and backword from the goal.

Subject :

Date :

Sudoku Puzzle



Consistency Check $h(n) \leq c(n,m) + h(m)$

$h(A) = 3$, $h(B) = 2$, $c(A,B) = 1$ (Actual path cost)

$3 \leq 1+2$ True

$h(B) = 2$, $h(C) = 2$; $c(B,C) = 3$

$2 \leq 3+2$ True

$h(B) = 2$, $h(D) = 1$, $c(B,D) = 2$

$2 \leq 2+1$ True

$h(C) = 2$, $h(D) = 1$, $c(C,D) = 2$

$2 \leq 2+1$ True

$h(D) = 1$, $h(G) = 0$, $c(D,G) = 3$

$1 \leq 3+0$ True

Consistency satisfied.

Subject: Date:

Admissible Graph

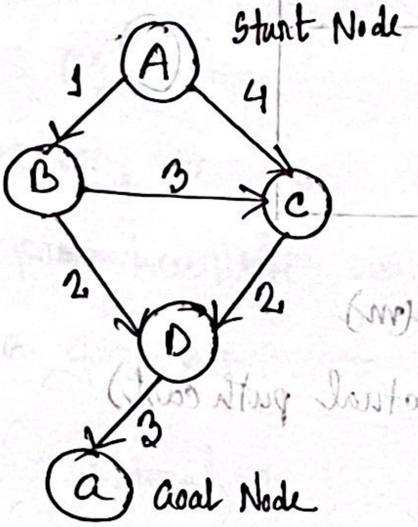
Conditions:

(a) Non-negative: $h(n)$ is admissible if $h(n) \geq 0$ for all nodes

(b) Never overestimate the actual cost. $h(n) \leq h^*(n)$

Underestimate $h(n) \leq h^*(n)$ | Here $h(n)$ = estimated value

Overestimate $h(n) \geq h^*(n)$ | $h^*(n)$ = actual value



Heuristic Table ($h(n)$)

equation: Minimum edge × steps to G

A → 3 steps and 1 edge = 3

B → 2 steps and 1 edge = 2

C → 2 steps and 1 edge = 2

D → 1 step and 1 edge = 1

G → 0 step and 0 edge = 0

Actual cost ($h^*(n)$)

A → C → D → G = 9

B → D → G = 5

C → D → G = 5

D → G = 3

G = 0

Admissible check

$h(n) = 3$, $h^*(n) = 9$

$h(n) \leq h^*(n)$ ✓

$h(n) = 2$, $h^*(n) = 5$

$h(n) \leq h^*(n)$ ✓

$h(n) = 2$, $h^*(n) = 5$

$h(n) \leq h^*(n)$ ✓

$h(n) = 1$, $h^*(n) = 3$

$h(n) \leq h^*(n)$ ✓

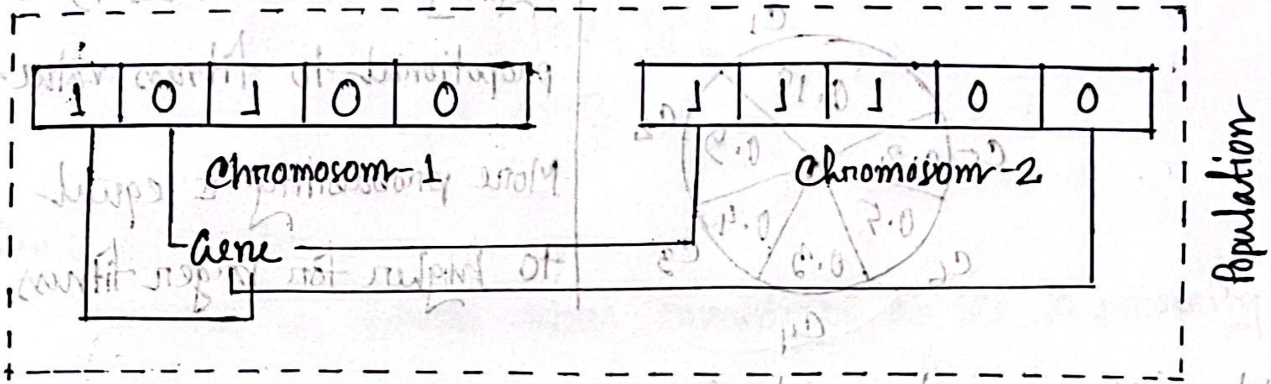
$h(n) = 0$, $h^*(n) = 0$

$h(n) \leq h^*(n)$ ✓

So, the graph is admissible

Subject : | Date :

Genetic Algorithm



A. Number of genes per chromosome

B. The codes value

C. The size of population per generation

D. Crossing over probabilities

E. Mutation probabilities

F. Termination criteria

Selection of Chromosome

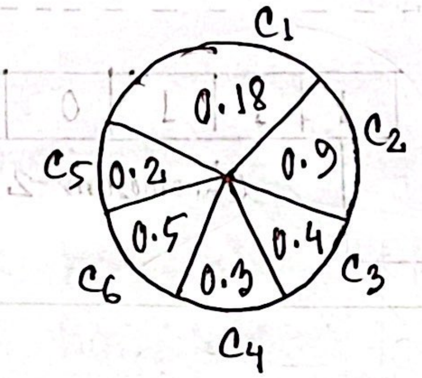
A. Roulette Wheel Selection

B. Rank Selection

C. Tournament Selection

Subject: Date:

Roulette Wheel Selection:



length of circumference is propotional to fitness value.

More probability is equal to higher for larger fitness.

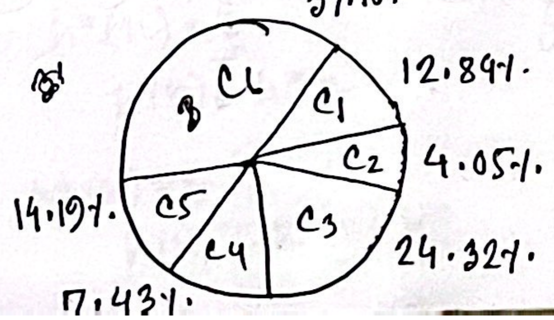
Determine parcentage of Roulette Wheel and Actual Count for six chromosomes with fitness value of 19, 6, 36, 11, 21 and 55.

Chromosome	Fitness	% of wheel	Probability	Expect Count	Actual Count
C_1	19	0.1284	12.84%	0.77	1
C_2	6	0.0405	4.05%	0.243	0
C_3	36	0.2432	24.32%	1.46	~~2~~ 1
C_4	11	0.0743	7.43%	0.44	1
C_5	21	0.1419	14.19%	0.85	1
C_6	55	0.3716	37.16%	2.23	2
	$\Sigma F = 148$				

Probality $\text{Count}_i = \frac{F_i}{\Sigma F}$ | For $C_1 = \frac{19}{148} = 0.1284$

$= 12.84\%$

Pie chart:



Subject: | Date:

Expected Count = Probabilities × Total Chromosome

For, $c_1 = 0.1284 \times 6$
$= 0.77$.

Rank Selection:

The less fitness values considered as the priority rank.

from the previous table; Selective Pressure [1.0, 2.0]

Chromosome	Fitness	Rank	New fitness	Previous %	New %
C_1	19	3	~~0.5~~ 0.9	12.84%	15%
C_2	6	1	~~0.75~~ 0.5	4.05%	8.33%
C_3	36	5	1.3	24.32%	21.667%
C_4	11	2	0.7	7.43%	11.667%
C_5	21	4	1.1	14.19%	18.334%
C_6	55	6	1.5	37.16%	25%
	$\Sigma F = 148$		$\Sigma nF = 6$		

$$Fit(X_R) = 2 - SP + 2(SP - 1)\frac{(Pos - 1)}{(Nind - 1)}$$

→ Selective pressure

→ position

→ Total number of chromosome

→ Constant value

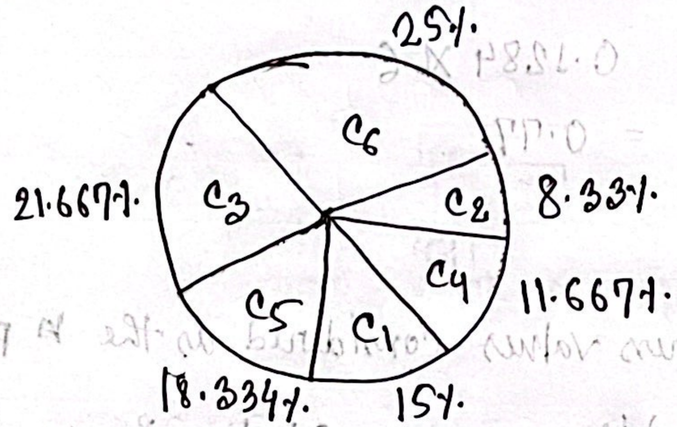
→ Rank Number (position)

Here,
$$SP = \frac{2.0 + 1.0}{2}$$
$$= 1.5$$

Selective pressure = [1.0, Nind − 1]

Subject: Date:

Pie chart:

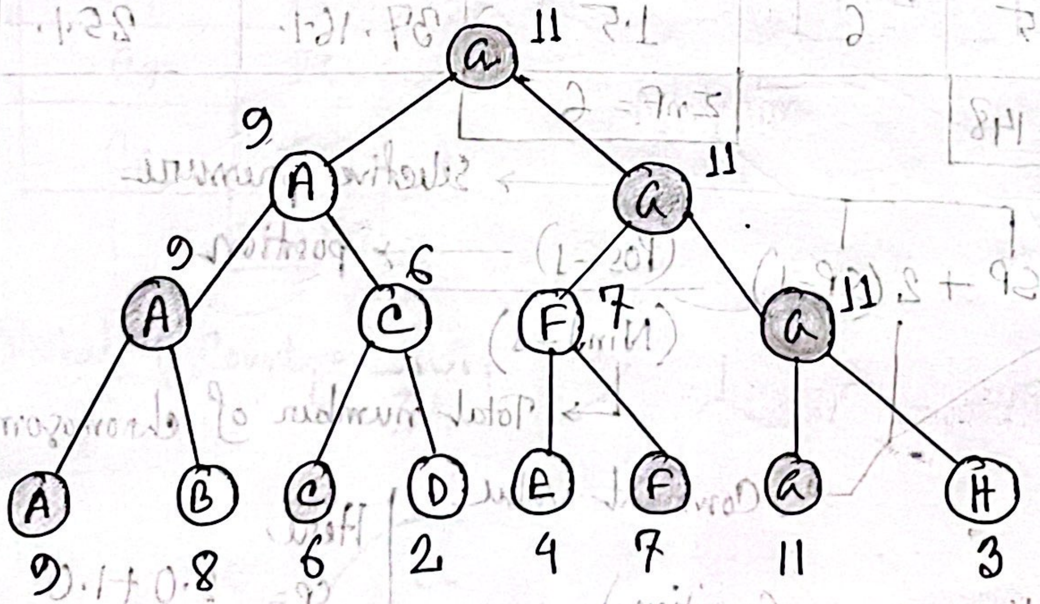


Tournament Selection:

(a) Select k individual from the population and performs a tournament among them.

(b) Select the best individual from the k individuals.

(c) Repeat the process 1 and 2 until we have the desired amount of population.



Subject: Date:

Crossover And Mutation

Crossover

0	1	0	0	1
1	0	1	1	0

0	1	0	1	0
1	0	1	0	1

(A) Single-site Crossover (B) Two-site Crossover (C) Crossover Mask

Single-site Crossover: (At point 5)

P_1	1	0	0	1	1	0	0	1	0
P_2	0	1	0	1	1	1	0	0	1

Strings Before mating

C_1	1	0	0	1	1	1	0	0	1
C_2	0	1	0	1	1	0	0	1	0

Strings After mating

Two-site Crossover: (At point 2 and 5)

P_1	1	0	1	0	1	0	0	1	0
P_2	0	1	0	1	0	1	0	0	1

C_1	1	0	0	1	0	0	0	1	0
C_2	0	1	1	0	1	1	0	0	1

Crossover Mask

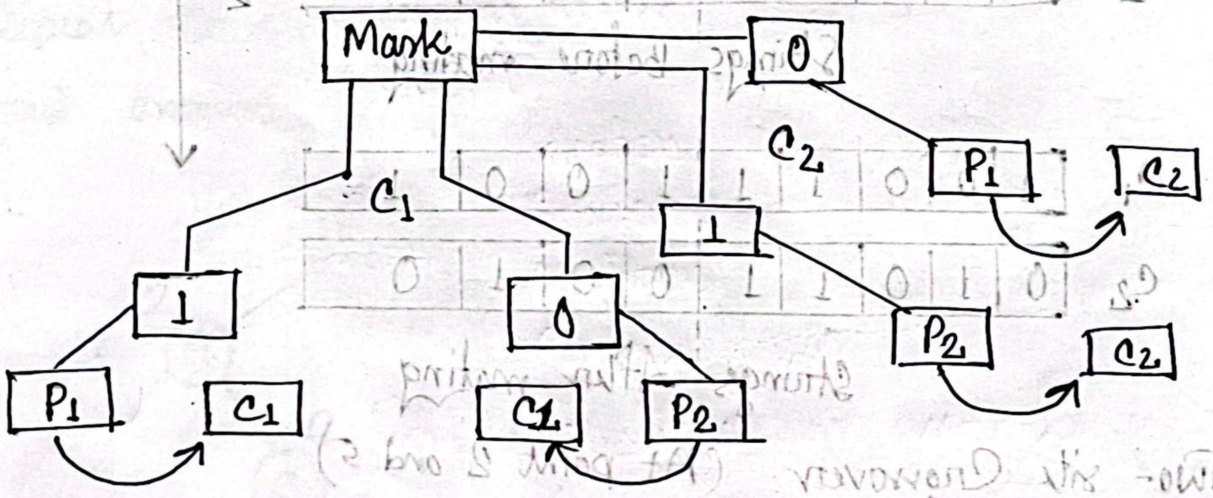
P_1	1	0	0	0	1	0	0	1	0
P_2	0	1	0	1	1	1	0	0	1

Strings Before mating

C_1	1	1	0	0	1	1	0	0	0
C_2	0	0	0	1	1	0	0	1	1

Strings After Mating

Mask	1	0	0	1	1	0	0	0	1



Subject: | Date:

Maximizing Function

Maximize the function $f(x) = x^2$, Where the range of x is $[0,31]$ and maximum number of generation is 3

Step 1: Initialization

00000, 00001, 00010, 00011, ... 11111

max number of generation = 3

Step 2: Selection (1st generation)

Chromosome	Encoding	Value of X	Fitness x^2
C_1	00101	5	25
C_2	10000	16	256
C_3	01001	9	81
C_4	11100	28	784
			$\Sigma f(x) = 1146$

Step 3: Crossover (2nd generation)

Chromosome	Encoding	Value of X	x^2	After Crossover
C_1	0010|1	4	16	00100
C_2	1000|0	17	289	10001
C_3	01|001	12	144	01100
C_4	11|100	25	625	11001
			$\Sigma f(x) = 1074$	

Subject: | Date:

Mutation (2nd generation)

Chromosome	Encoding	Values of X^2	$f(x) = x$	After Mutation
C_1	00100	16	4	00100
C_2	10001	441	21	10101
C_3	01100	144	12	01100
C_4	11001	729	27	11011
			$\Sigma f(x) = 1330$	

Pseudo-Code of Genetic Algorithm:

```
START
    Generate the initial population
    Compute fitness
    REPEAT
        Selection
        Crossover
        Mutation
        Compute fitness
    UNTIL population has over converged
STOP
```

Subject :   Date :

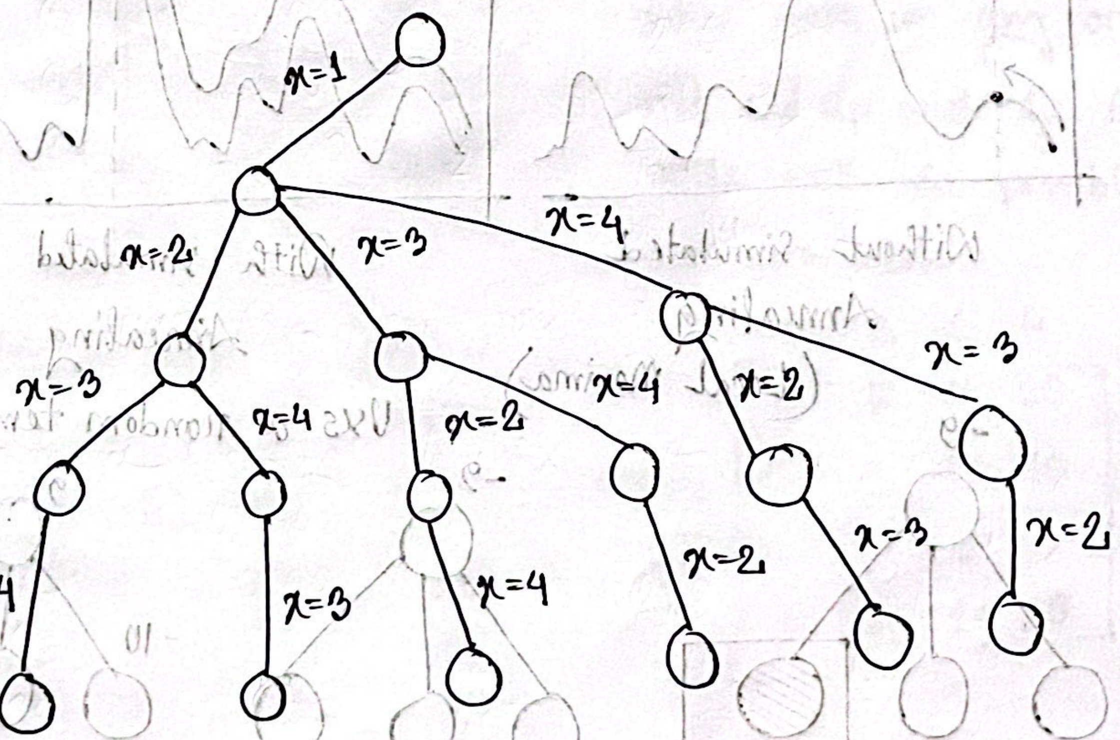
Queen Problem using Backtracking

	1	2	3	4
1	Q_1			
2		Q_2		
3			Q_3	
4				Q_4

4 Queens

$Q_1 \mid Q_2 \mid Q_3 \mid Q_4$

Queens can not be in same row and same columns.

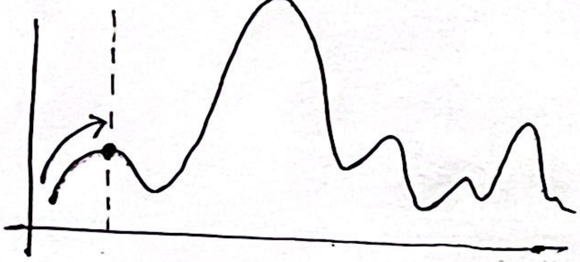


Subject: | Date:

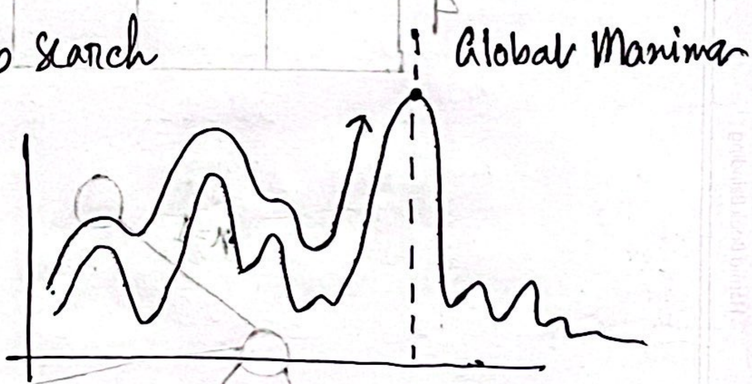
Local Search

(a) Not concerned with path
(b) Solution itself matters
(c) Not concerned with perfect solution
(d) less memory required as its not using Backtracking
(e) less time required
(f) Applicable for large size of graph

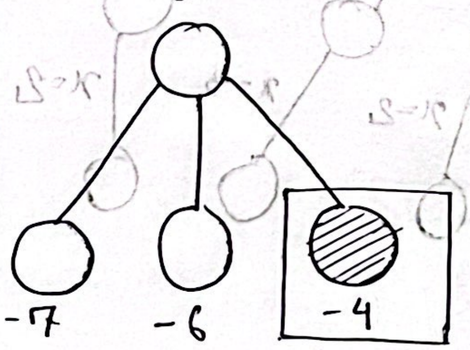
Hill Climb search



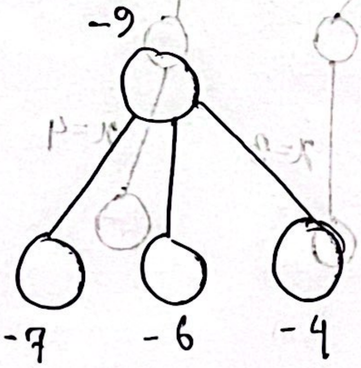
Without simulated Annealing (Local Maxima)



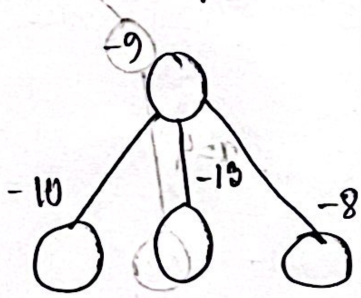
With simulated Annealing
Uses a random temperature



Hill climb Steepest
select the maximum



Hill climb Stochastic
select a random node



Hill Climb First choice
select the first occurrence of minimum

Subject:

Date:

Simulated Annekling

It allowes downWard step also.

Advantages:

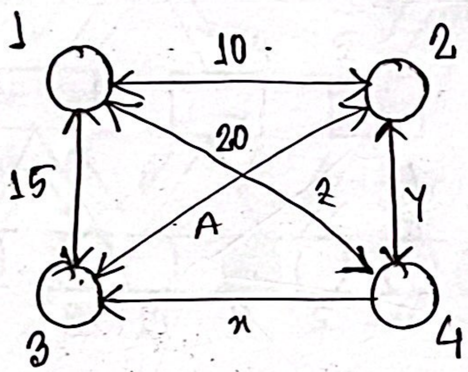
Always gives optimal Solutions

Disadvantages:

Slow process

Cann't tell if an optimal solution is found.

TSP (Travelling Salesman Problem)



Start from node 1, travel the entire path in any order and end up with the same node with the possible minimum cost.

ΔE = New Cost - Old Cost

if $\Delta E < 0$, accept

if $\Delta E > 0$, find the probability

$$P = e^{-\Delta E/T}.$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & \multicolumn{4}{c}{} \\ \end{array}$$

	1	2	3	4
1	0	10	15	20
2	5	0	25	10
3	15	30	0	5
4	15	10	20	0

Higher temperature allows Worse Solutions to be accepted. helping escap local optima. When the temperature is high, P is close to Worse solutions.

Subject :

Date :

Question Pattern

Pobular Problem

Travelling Salesman problem
- Local Search Algorithm
- Hill Climbing Algorithm

N- Queens Problem
- Hill Climbing
- Local Search

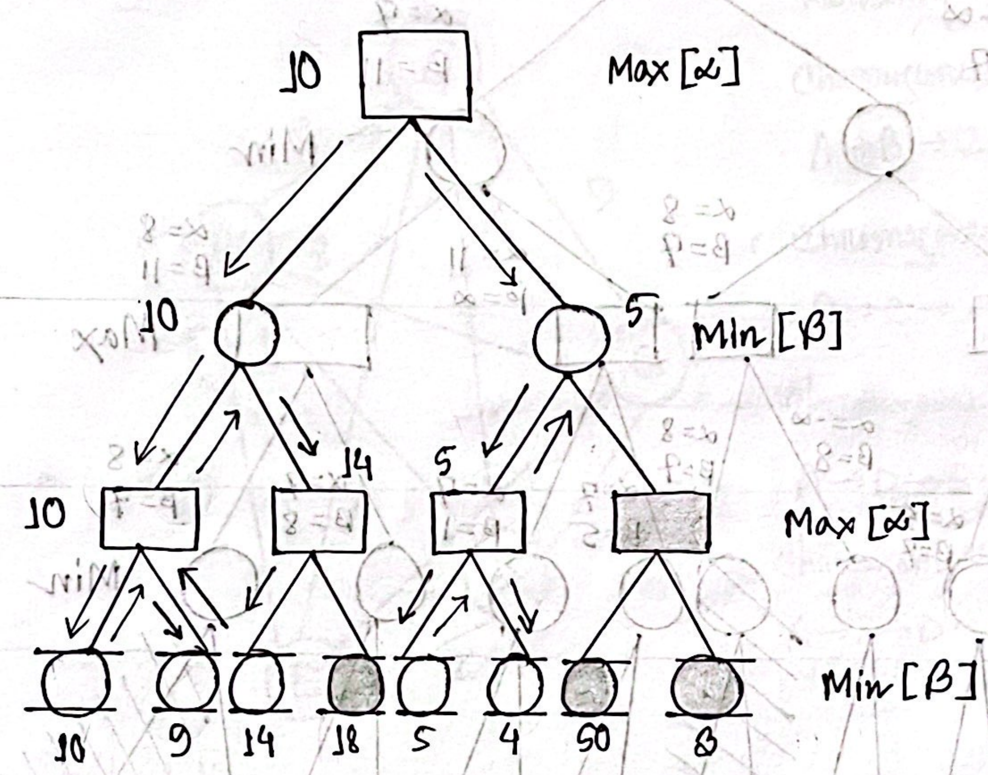
8 - Putt Puzzle / 15- Puzzle Problem

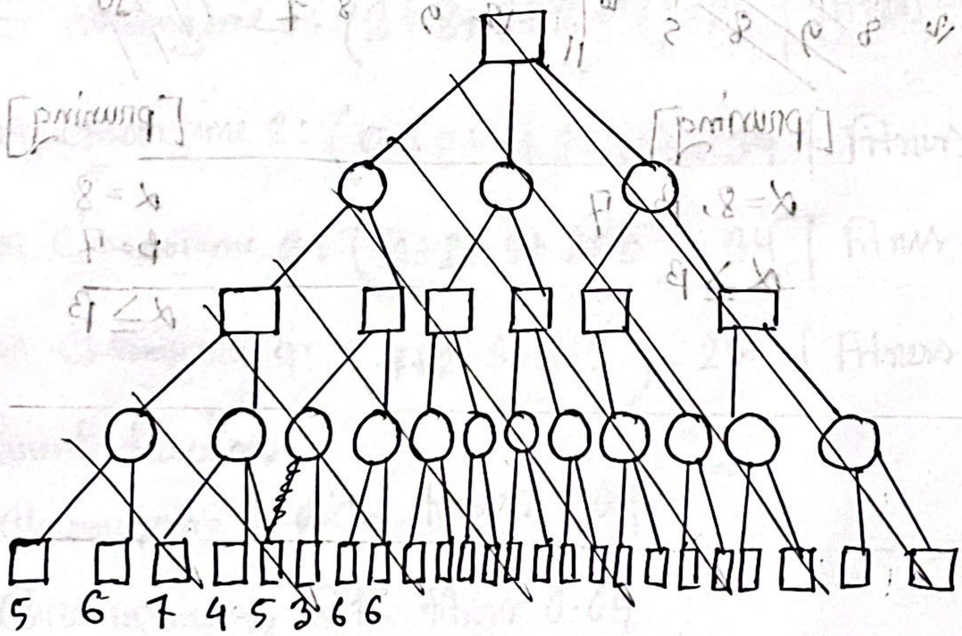
Subject: | Date:

MiniMax using Alpha-Beta Pruning

Max contains Alpha

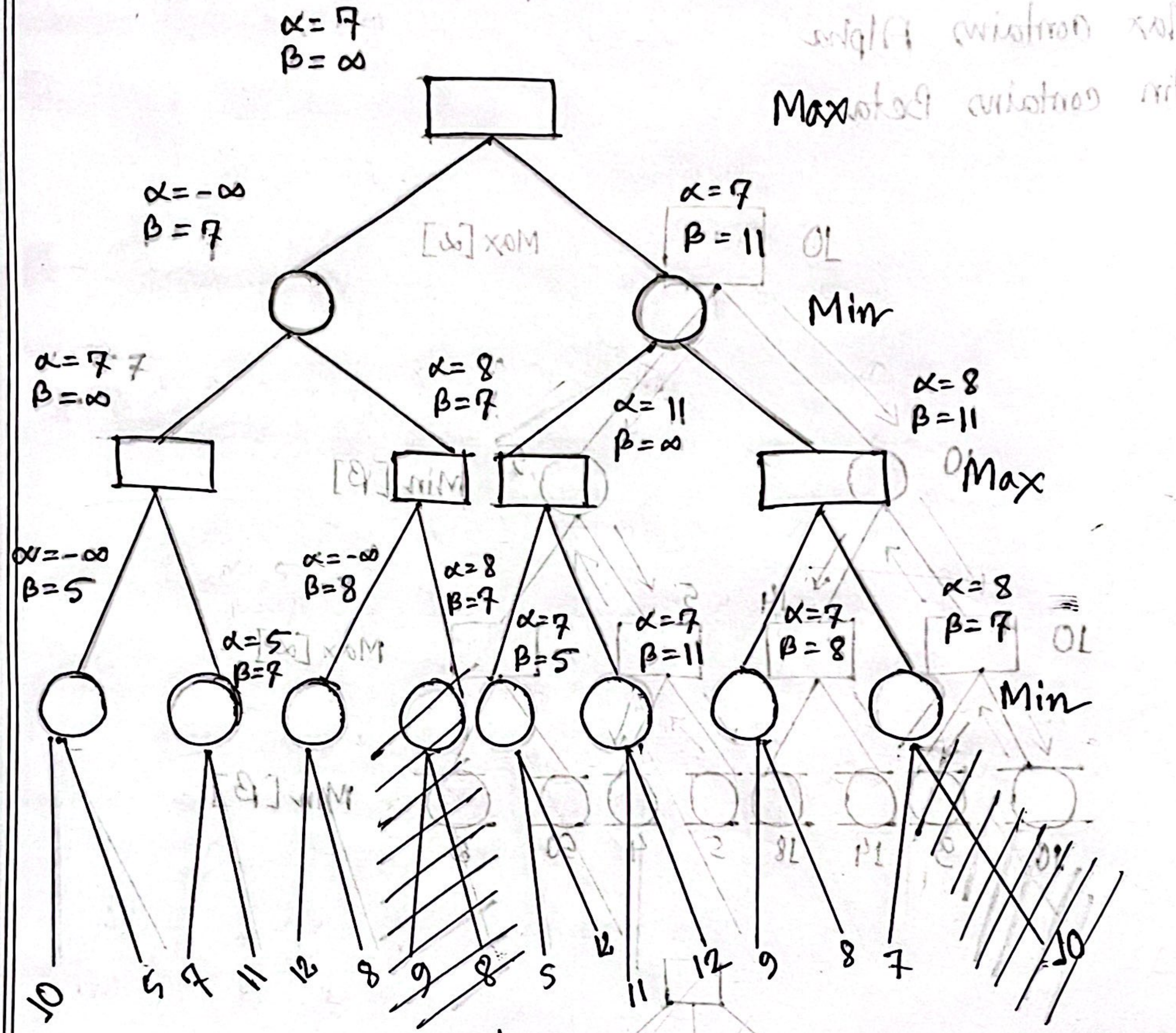
Min contains Beta





Subject : Date :

Minimax Alpha Beta pruning Simulation



[pruning]

$\alpha = 8, \beta = 7$

$\alpha \geq \beta$

[pruning]

$\alpha = 8$

$\beta = 7$

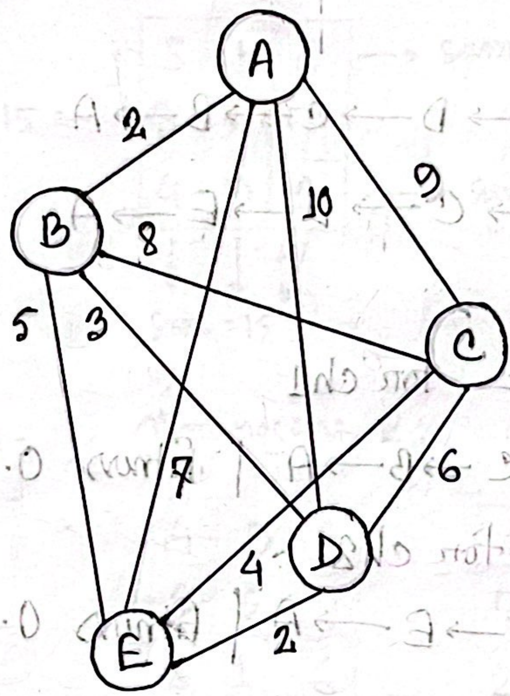
$\alpha \geq \beta$

Subject: Date:

Genetic Algorithm Problem Solving AI

Problem 1:

Travelling Salesman Problem



Randomly Selection:

Chromosome 1:

A → B → C → D → E → A

Chromosome 2:

A → C → B → E → D → A

Chromosome 3:

A → D → E → B → C → A

Chromosome 4:

A → E → D → C → B → A

Fitness Calculation:

For chromosome 1: $(2+8+6+2+7)=25$ | Fitness $= \frac{1}{25} = 0.04$

For chromosome 2: $(9+8+5+2+10) = 34$ | Fitness $= 1/34 = 0.0294$

For chromosome 3: $(10+2+5+8+9) = 34$ | Fitness $= 1/34 = 0.0294$

For chromosome 4: $(7+2+6+8+2) = 25$ | Fitness $= 1/25 = 0.04$

Parent Selection:

Chromosome 1 with fitness 0.04

Chromosome 4 with fitness 0.04

Subject:

Date:

Crossover:

chromosome 1: A → B → C → D → E → A

chromosome 2: A → E → D → C → B → A

child 1: A → B → D → C → B → A

child 2: A → E → C → D → E → A

Mutation:

Swap B with E for ch1

A → E → D → C → B → A | fitness 0.040

Swap C with B for ch2

A → E → B → D → E → A | fitness 0.0416

The answer A → E → B → D → E → A

Distance: 24

fitness: 0.0416

Subject: Date:

Problem 2:

Rubik's Cube Problem

given that. The final state is

8	1	6
3	5	7
4	9	2

$\rightarrow$ sum = 15

sum 15 $\leftarrow$

$\rightarrow$ sum 15

sum = 15

Chromosome 1:

2	7	6
9	5	1
4	3	8

Chromosome 2:

4	9	2
3	5	7
8	1	6

Chromosome 3:

6	1	8
7	5	3
2	9	4

Chromosome 4:

9	3	5
4	1	8
7	2	6

Fitness calculation:

$$\text{fitness} = \frac{1}{1 + |(15 - \text{row sum})| + |(15 - \text{column sum})| + |(15 - \text{diagonal sum})|}$$

Chromosome 1:

Rows: $|15-15| + |15-15| + |15-15| = 0$

Column: $|15-15| + |15-15| + |15-15| = 0$

Diagonal: $|15-15| + |15-15| = 0$

Fitness $= \frac{1}{(0+1)} = 1$

Subject : Date :

Chromosome 2:
Fitness : 1

Chromosome 3:
Fitness : 1.0

Chromosome 4:
Rows : $|15-17| + |15-13| + |15-15| = 4$
Columns: $|15-16| + |15-13| + |15-16| = 6$
Diagonals : $|15-16| + |15-17| = 3$
Fitness : 0.0833

Parent 1 : Chromosome 2

Parent 2 : Chromosome 1

ch2

$$\begin{matrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{matrix} \longleftrightarrow \begin{matrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{matrix}$$

ch1

Child 1:

$$\begin{matrix} 4 & 9 & 2 \\ 9 & 5 & 1 \\ 8 & 1 & 6 \end{matrix}$$

swap 9 and 3

Child 2:

$$\begin{matrix} 2 & 7 & 6 \\ 3 & 5 & 7 \\ 4 & 3 & 8 \end{matrix}$$

swap 7 and 9

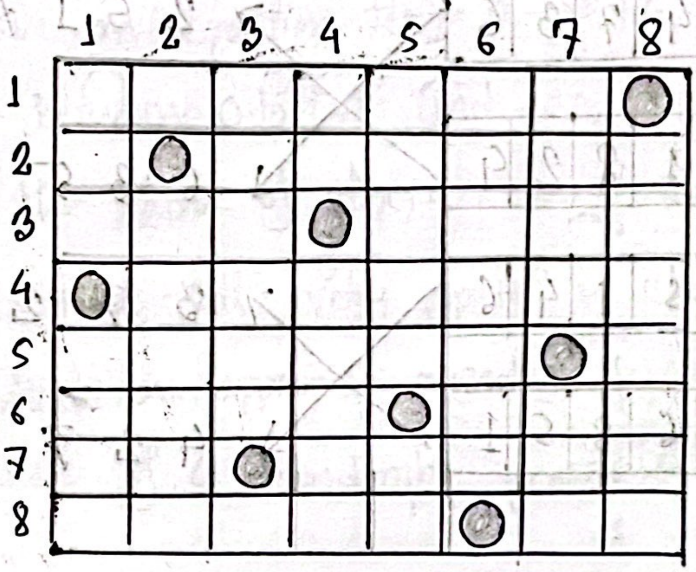
Mutation

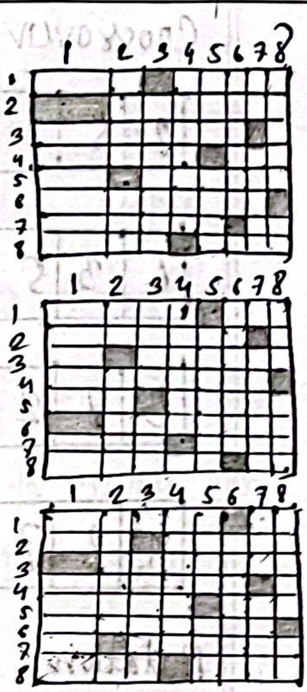
$$\begin{matrix} 4 & 3 & 2 \\ 9 & 5 & 1 \\ 8 & 1 & 6 \end{matrix} \qquad \begin{matrix} 2 & 9 & 6 \\ 3 & 5 & 7 \\ 4 & 3 & 8 \end{matrix}$$

Subject: Date:

Problem 3:

8 Queen Problem





Representation:

[4, 2, 7, 3, 6, 8, 5, 1]

Here, index = column
Value = row

Chromosome 1: [4, 2, 7, 3, 6, 8, 5, 1]

Chromosome 2: [2, 5, 1, 8, 4, 7, 3, 6]

Chromosome 3: [6, 3, 5, 7, 1, 8, 2, 4]

Chromosome 4: [3, 7, 2, 8, 5, 1, 4, 6]

fitness:

$q_1 = 6$ $\quad \frac{\text{total}}{56}$
$q_2 = 8$
$q_3 = 6$ $\quad \Sigma_T = 240$
$q_4 = 8$
$q_5 = 8$ $\therefore$ fitness
$q_6 = 6$ $\quad 23.33\%$
$q_7 = 8$
$q_8 = 6$

$q_1 = 8$ $\quad \frac{\text{total}}{56}$
$q_2 = 8$
$q_3 = 6$ $\quad \Sigma_T = 240$
$q_4 = 8$
$q_5 = 6$ $\therefore$ fitness
$q_6 = 6$ $\quad 23.33\%$
$q_7 = 8$
$q_8 = 6$

$q_1 = 8$ $\quad \frac{\text{total}}{64}$
$q_2 = 8$
$q_3 = 8$ $\quad \Sigma T = 240$
$q_4 = 8$
$q_5 = 8$ $\therefore$ fitness:
$q_6 = 8$ $\quad 35\%$
$q_7 = 8$
$q_8 = 8$

$q_1 = 8$ $\quad \frac{\text{total}}{64}$
$q_2 = 8$
$q_3 = 8$ $\quad \Sigma_T = 240$
$q_4 = 8$
$q_5 = 8$ $\quad$ Fitness:
$q_6 = 8$ $\quad 35\%$
$q_7 = 8$
$q_8 = 8$

Subject :

Date :

Crossover :

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

2 5 1 7 1 8 2 4

6 3 5 8 4 7 3 6

3 7 2 3 6 8 5 1

4 2 7 8 5 1 4 6

Mutation :

2 5 1 8 1 8 2 4

6 3 5 8 4 4 3 6

3 7 2 1 6 8 5 1

4 2 1 8 8 1 4 6

Subject: Date:

Mid Question (Genetic Algorithm)

1. Suppose you have an equation $f(x) = x^2 - 5x + 6$. Assume x can be any number between 0 to 15. Find an appropriate value of x such that the value of $f(x) = 0$ using Genetic Algorithm

(a) Consider the fact that every population chromosome will have 4 genes. Illustrate an appropriate encoding system to create an initial population of 4 randomly generated chromosome.

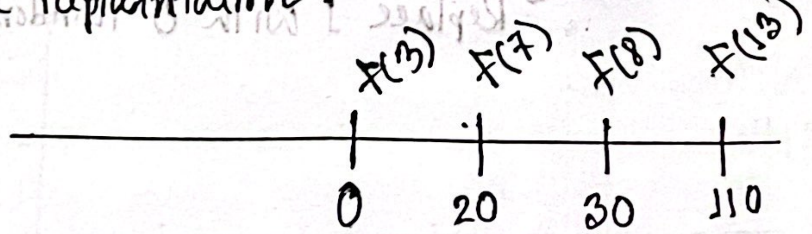
$$f(x) = x^2 - 5x + 6$$

BCD representation: (Encoding system)

8	4	2	1	
0	1	1	1	chromosome 1
0	0	1	1	chromosome 2
1	0	0	0	chromosome 3
1	1	0	1	chromosome 4

(b) Using an appropriate fitness function deduce the 2 fittest chromosome and perform a single pointer crossover from the middle to create two offspring.

Number Line representation:



equal to zero ← | ↳ close to zero

Subject: Date:

parent 1 : fitness ∞

0	0	1	1

parent 2 : fitness 0.05

0	1	1	1

Fitness calculation

$$\text{fitness} = \frac{1}{f(x)}$$

for $f(3)$, fitness $= \infty$

for $f(7)$, fitness $= 0.05$

for $f(8)$, fitness $= 0.0333$

for $f(13)$, fitness $= 9.09 \times 10^{-3}$

Single point crossover:

Assume, point = 2

0	0	1	1
0	1	1	1

→

0	0	1	1
0	1	1	1

ⓒ Explain how you can mutate the offspring.

Mutation

Replace 0 with 1 randomly

1	0	1	1

$$\text{fitness}: \frac{1}{f(11)} = 0.01388$$

0	0	1	1

fitness: ~~0.05~~ ∞

Replace 1 with 0 randomly

Subject: Date:

(d) Explain your opinion on whether Genetic Algorithm can be treated as a class of Local Search Algorithms or not.

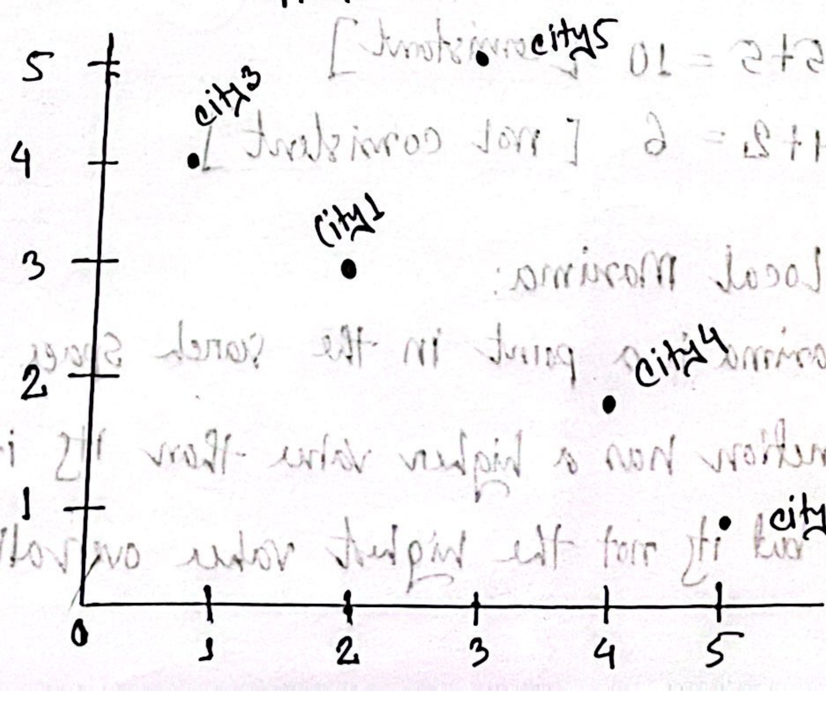
GA is better to be considered as global search algorithm for its diversity and mutation nature.

2. Ruhan is a salesperson. His point is to find the optimal path that visit all cities at once and return to the starting point.

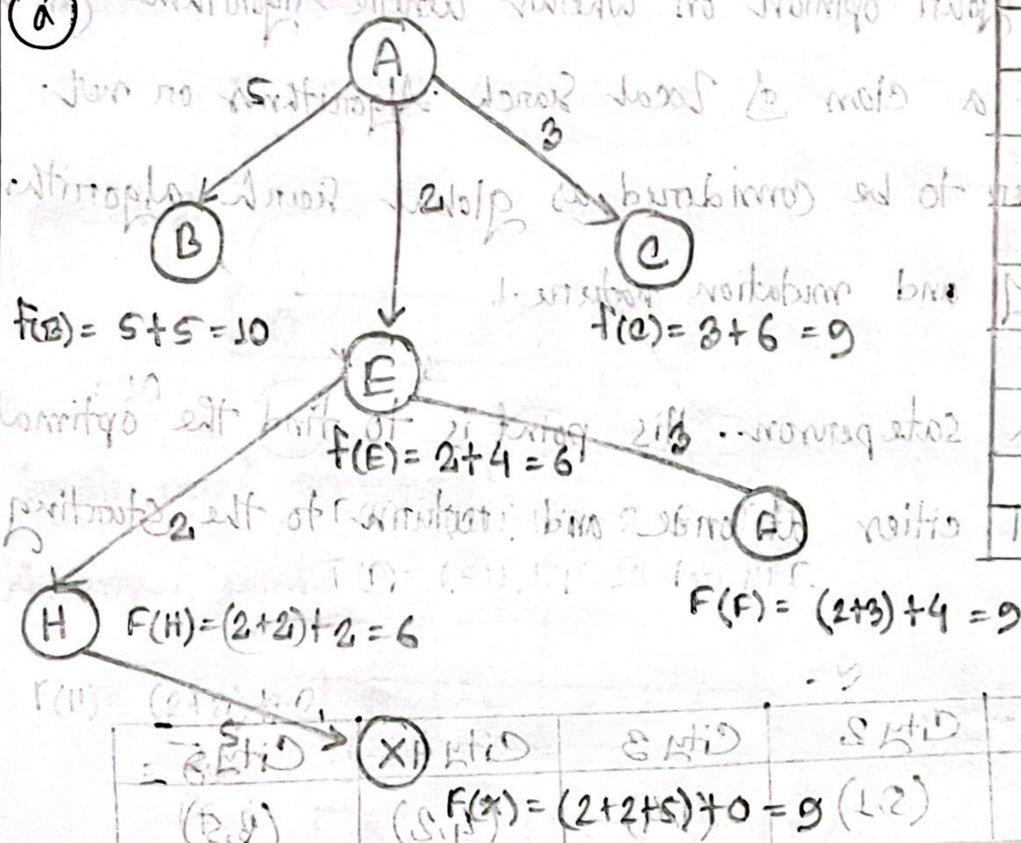
City 1	City 2	City 3	City 4	City 5
(2,3)	(5,1)	(1,4)	(4,2)	(3,5)

Use genetic algorithm to find solutions.

(a) Encode the problem and create four parent chromosomes. Then determine an appropriate fitness function and choose parents.



3. ⓐ



Node	h-value
A	7
B	5
C	6
D	3
E	4
F	4
G	2
H	2
X	0

ⓑ Check the Consistancy.

Hueristic value of Node 1 $\leq$ Hueristic value of node 2 + path cost

Node A :

$7 \leq 5+5 = 10$ [consistant]

$7 \leq 4+2 = 6$ [not consistent].

ⓒ Define Local Maxima:

A local maxima is a point in the search space where the objective function has a higher value than its immediate neighbours but its not the highest value overall.

Subject: Date:

Example: Hill Climbing. it Will stuck at local maxima but there will be a better solution elsewhere.

(2) Define Local minima:

A local minima is a point where the objective function has a lower value than its imediate neighbour. but not the lowest at all.

Hill climbing: often stuck in local maxima

Simulated Annealing: Escape local maxima/minima

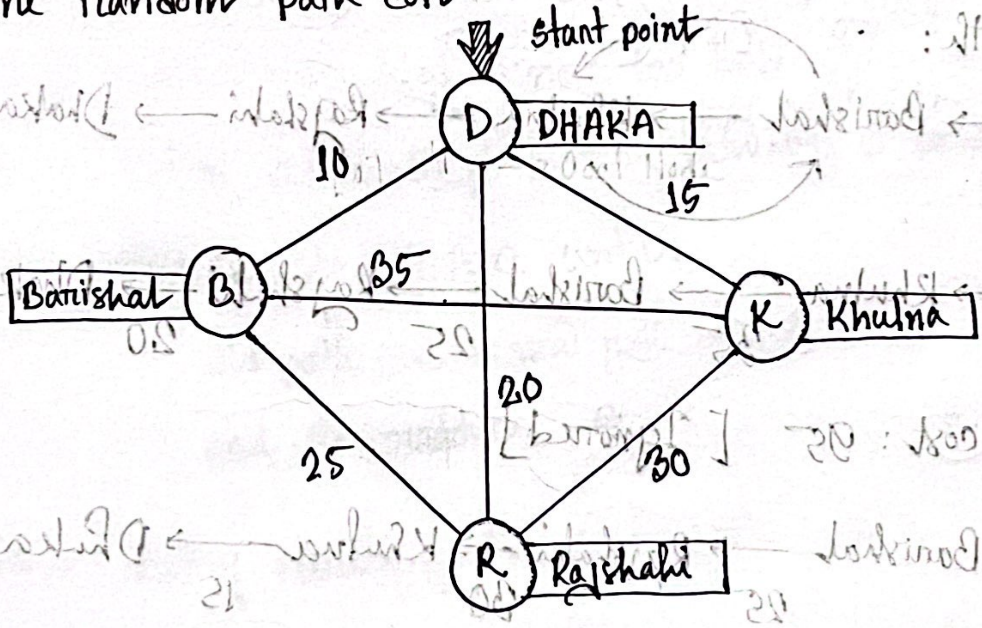
GA: Avoid local traps and find global optima

Subject: Date:

Question 2:

Travelling Salesman Problem is a path travelling problem where the algorithm finds that path which is the shortest among all other paths possible from start point to start point after visiting every other nodes exactly one time.

To solve the problem lets assume that we have 4 cities and they are interconnected as mesh topology structure with some random path cost.



Approch:

ⓐ Select any initial Route:

$$D \xrightarrow{10} B \xrightarrow{35} K \xrightarrow{30} R \xrightarrow{20} D$$

Total distance = 95 unit

Subject: Date:

(6) Generate Random path :

We will generate random path by swapping any two cities.

Then we will calculate the path cost.

If the total cost $\geq$ previous cost :

Ignore !

If the total cost $<$ previous cost :

Accepted !

previous path :

Dhaka $\longrightarrow$ Barishal $\longrightarrow$ Khulna $\longrightarrow$ Rajshahi $\longrightarrow$ Dhaka

Dhaka $\xrightarrow[15]{}$ Khulna $\xrightarrow[35]{}$ Barishal $\xrightarrow[25]{}$ Rajshahi $\xrightarrow[20]{}$ Dhaka

Total path cost : 95 [Ignored]

Dhaka $\xrightarrow[10]{}$ Barishal $\xrightarrow[25]{}$ Rajshahi $\xrightarrow[30]{}$ Khulna $\xrightarrow[15]{}$ Dhaka

Total path cost : 80 [Accepted]

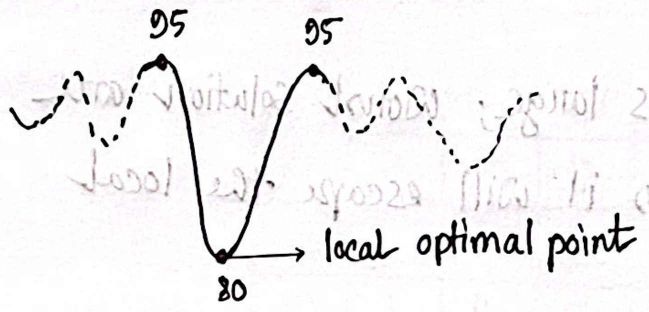
Dhaka $\xrightarrow[20]{}$ Rajshahi $\xrightarrow[25]{}$ Barishal $\xrightarrow[35]{}$ Khulna $\xrightarrow[15]{}$ Dhaka

Total path cost : 95 [Ignored and Stop searching]

Return : 80 , Path : D $\longrightarrow$ B $\longrightarrow$ R $\longrightarrow$ K $\longrightarrow$ D

Subject: Date:

Visual Representation:



Question 3:

Step A: Initialize a random path and find the path cost.

step B: Initialize a random temperature. The selected temperature neither too high nor too low.

step C: Define a cooling factor between 0 and 1.

step D: Repeat step A, B, C until:

old path cost − new path cost < 0

and temperature value is very small.

Step E: Every iteration / Function call

compute $\Delta f = f(S_{new}) - f(S_{old})$

if $\Delta f < 0$; ACCEPTED

else:

ACCEPTED with propability $= e^{\frac{-\Delta E}{T}}$

$T = \alpha$ (cooling factor) $* T$

Subject: Date:

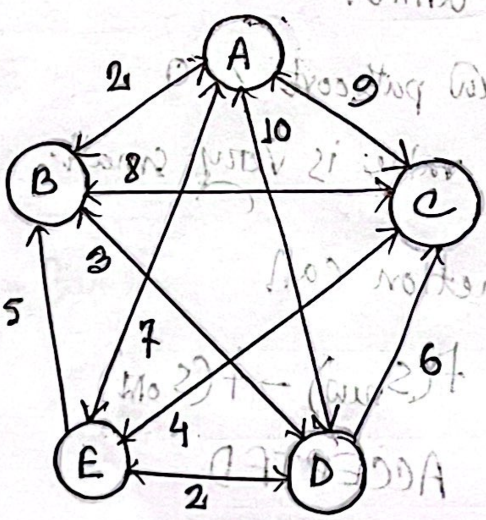
Step F: Every iteration the temperature (T) will decrease.

If probabity is large; worst solution are accepted means it will escape the local minima.

Step G: When the T is very low to make the next search, The program will stop finding.

Step H: Return the path with path cost.

Question 4:



Subject: Date:

Iteration 1:

Initialization of chromosome:

chromosome 1:

A	B	C	D	E	A

Chromosome 2:

A	C	B	E	D	A

chromosome 3:

A	D	E	B	C	A

chromosome 4:

A	E	D	C	B	A

fitness Calculations:

for chromosome 1: $(2+8+6+2+7)=25$

fitness: $\frac{1}{25} = 0.04$ [Highest fitness]

for chromosome 2: $(9+8+5+2+10) = 34$

fitness: $\frac{1}{34} = 0.0294$

for chromosome 3: $(10+2+5+8+9) = 34$

fitness: $\frac{1}{34} = 0.0294$

for chromosome 4: $(7+2+6+8+2) = 25$

fitness: $\frac{1}{25} = 0.04$ [Highest fitness]

Subject: | Date:

Crossover:

chromosome 1: A → B → C → D → E → A

chromosome 2: A → E → D → C → B → A

Child 1: A → B → D → C → B → A

Child 2: A → E → C → D → E → A

Mutation:

Swap B with E for child 1

Swap E with B for child 2

Child 1: A → B → D → C → E → A

Child 2:

A → E → C → D → B → A

Iteration 2:

Initialize chromosome:

chromosome 1:

A	B	D	C	E	A

chromosome 2:

A	E	C	D	B	A

Child 1 and Child 2 in previous iteration

Chromosome 3:

A	C	B	E	D	A

Chromosome 4:

A	D	E	B	C	A

Subject: Date:

Fitness Calculations:

For chromosome 1: $(2+3+6+4+7) = 22$

fitness $= 1/22 = 0.04545$ [Highest fitness]

For chromosome 2: $(7+4+6+3+2) = 22$

Fitness $= 1/22 = 0.04545$ [Highest fitness]

For chromosome 3: $(9+8+5+2+10) = 34$

fitness $= 0.029411$

For chromosome 4: $(10+2+5+8+9) = 34$

Fitness $= 1/34 = 0.029411$

Crossover:

chromosome 1: A → B → D → C → E → A

Chromosome 2: A → E → C → D → B → A

Child 1: A → B → D → D → B → A

child 2: A → E → C → C → E → A

Mutation:

swap D with C for child 1

Swap C with B for child 2

Subject:

Date:

Child 1: A ⟶ B ⟶ C ⟶ D ⟶ B ⟶ A

Child 2: A ⟶ E ⟶ B ⟶ C ⟶ E ⟶ A

Iteration 3:

Initialize chromosome:

Chromosome 1:

A	B	C	D	B	A

Chromosome 2:

A	E	B	C	E	A

Chromosome 3:

A	C	B	E	D	A

Chromosome 4:

A	D	E	B	C	A

Fitness Calculations:

For chromosome 1: $(2+8+6+3+2) = 21$

Fitness $= 1/21 = 0.04761$ [Highest fitness]

For chromosome 2: $(7+5+8+4+7) = 31$

Fitness $= 1/31 = 0.03225$ [Highest fitness]

For chromosome 3: $(9+8+5+2+10) = 34$

Fitness $= 1/34 = 0.029411$

For chromosome 4: $(10+2+5+8+9) = 34$

Fitness $= 1/34 = 0.029411$

Subject:

Date:

Crossover:

Chromosome 1: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow B \rightarrow A$

Chromosome 2: $A \rightarrow E \rightarrow B \rightarrow C \rightarrow E \rightarrow A$

Child 1: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow A$

Child 2: $A \rightarrow E \rightarrow B \rightarrow C \rightarrow B \rightarrow A$

Mutations:

Swap B with D in child 1

Swap B with D in child 2

Child 1: $A \rightarrow D \rightarrow C \rightarrow D \rightarrow E \rightarrow A$

$$\text{fitness} = 1/(10+6+6+2+7) = 0.032258$$

Child 2: ~~$A \rightarrow E \rightarrow B \rightarrow C \rightarrow D \rightarrow A$~~

~~$\text{fitness} = 1/(7+5+8+6+10)$~~

$A \rightarrow E \rightarrow D \rightarrow C \rightarrow B \rightarrow A$

$$\text{fitness}: \frac{1}{(7+2+6+8+2)} = 1/25 = 0.04$$

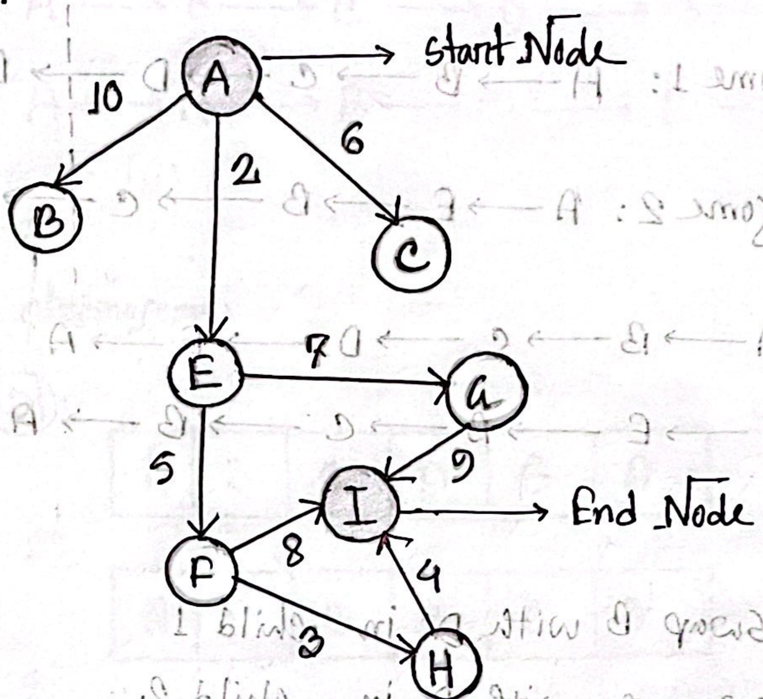
Final Path: $A \rightarrow E \rightarrow D \rightarrow C \rightarrow B \rightarrow A$

Cost: 25, fitness: 0.04.

Subject: Date:

Question 1:

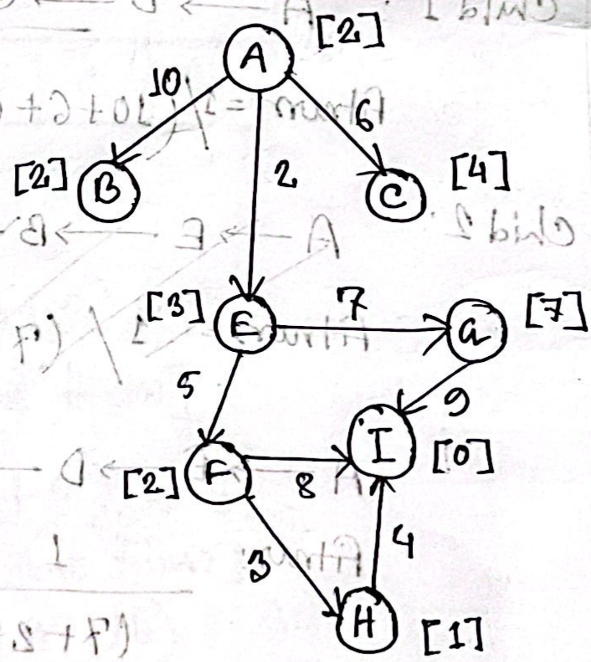
[A, B]



B.

C.

Node	Heuristic Value
A	2
B	2
C	4
~~D~~ I	0
E	3
F	2
G	7
H	1



Subject: Date:

D.

Start Node = A

End Node = I

Find the lowest / shortest path from A to I

path 1: $A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{8} I$, cost = 15

path 2: $A \xrightarrow{2} E \xrightarrow{7} G \xrightarrow{9} I$, cost = 18

path 3: $A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{3} H \xrightarrow{4} I$, cost = 14

$\therefore$ The shortest path: $A \longrightarrow E \longrightarrow F \longrightarrow H \longrightarrow I$

The graph is admissible if Node A, E, F, H and I are admissible simultaneously.

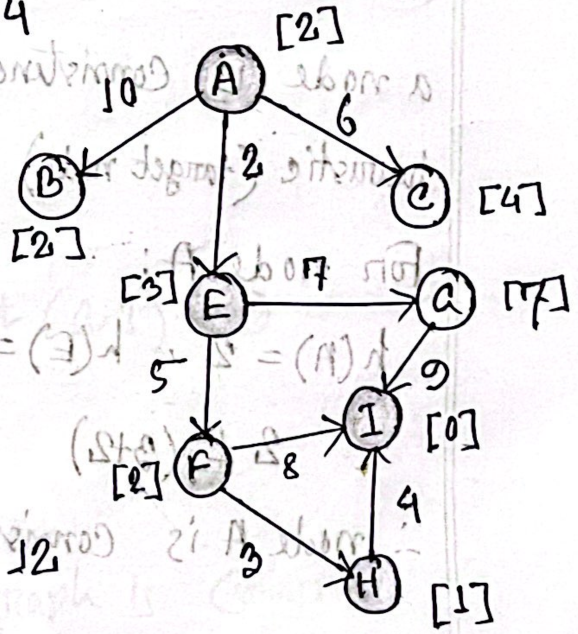
For checking admissibility,

Admissible if, heuristic value of $(n) \leq$ actual path cost from (n)

For Node A:

$h(A) = 2$ and $h^*(A) = (2+5+3+4) = 14$

$\therefore h(A) \leq h^*(A)$



For Node B:

$h(B) = 2$ [edge node]

For Node C:

$h(C) = 4$ [edge node]

For Node E:

$h(E) = 3$ and $h^*(E) = (5+3+4) = 12$

$\therefore h(E) \leq h^*(E)$

Subject: Date:

For Node F:

$h(F) = 2$ and $h^*(F) = (3+4) = 7$

$\therefore h(F) \le h^*(F)$

For Node H:

$h(H) = 1$ and $h^*(H) = 4$

$\therefore h(H) \le h^*(H)$

For Node I:

$h(I) = 0$ and $h^*(I) = 0$

$\therefore h(I) \le h^*(I)$

For Node G:

$h(G) = 7$ and $h^*(G) = 9$

$\therefore h(G) \le h^*(G)$

$\therefore$ The graph is admissible. [proved]

E. The shortest path from Node A to Node I is

$A \longrightarrow E \longrightarrow F \longrightarrow H \longrightarrow I$

a node is Consistence if

heuristic (target node) $\le$ heuristic (next node) + path cost (m, n)

For node A:

$h(A) = 2$, $h(E) = 3$, path cost $(A, E) = 2$

$2 \le (3+2)$

$\therefore$ node A is consistance.

Subject: | Date:

For node E:

$h(E) = 3$, $h(F) = 2$, path cost $(E,F) = 5$

$h(G) = 7$, path cost $(E,G) = 7$

$3 \leq (2+5)$ and $3 \leq (7+7)$

$\therefore$ node E is consistance.

For node F:

$h(F) = 2$, $h(I) = 0$, path cost $(F,I) = 8$

$h(H) = 1$, path cost $(F,H) = 3$

$2 \leq (0+8)$ and $2 \leq (1+3)$

$\therefore$ Node F is consistance.

For node H:

$h(H) = 1$, $h(I) = 0$, path cost $(I,H) = 4$

$1 \leq (0+4)$

$\therefore$ node H is consistance.

For node G:

$h(G) = 7$, $h(I) = 0$, path cost $(G, I) = 9$

$0 \leq (7+9)$

$\therefore$ node G is consistance.

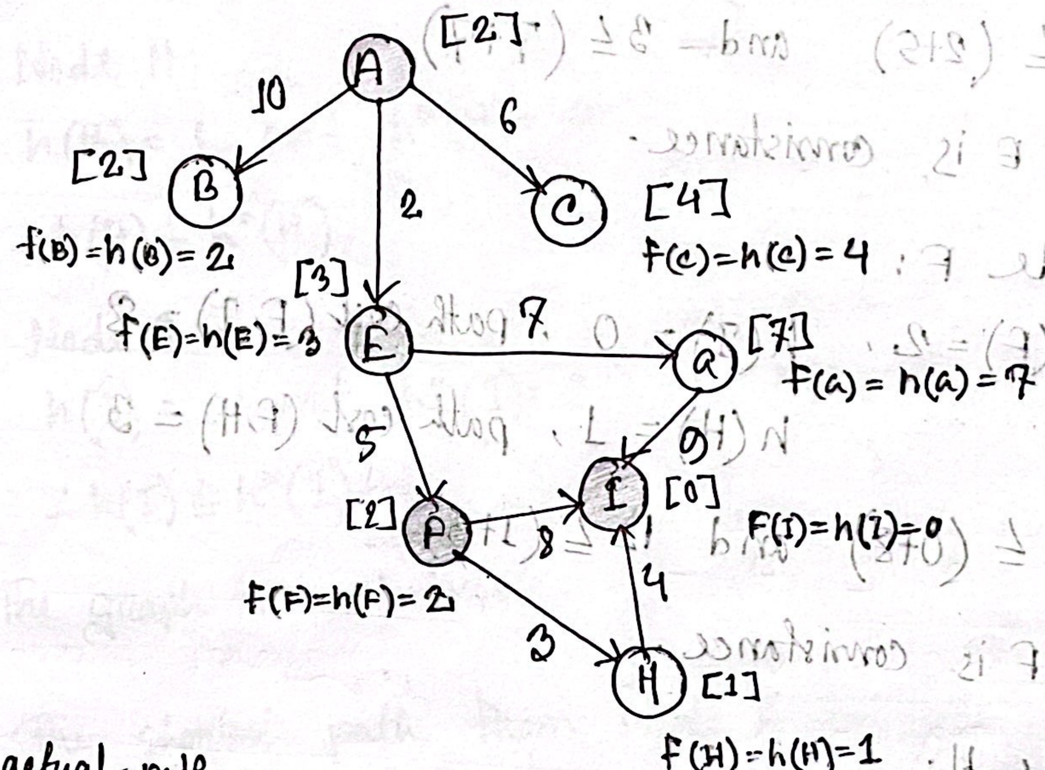
As all nodes are consistance. The graph is Consistan.

Subject: | Date:

F.

Technique 1:

Greedy Best First Search



The actual path:

$A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{3} H \xrightarrow{4} I$

Total path cost = 14

GBFS path:

$A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{8} I$

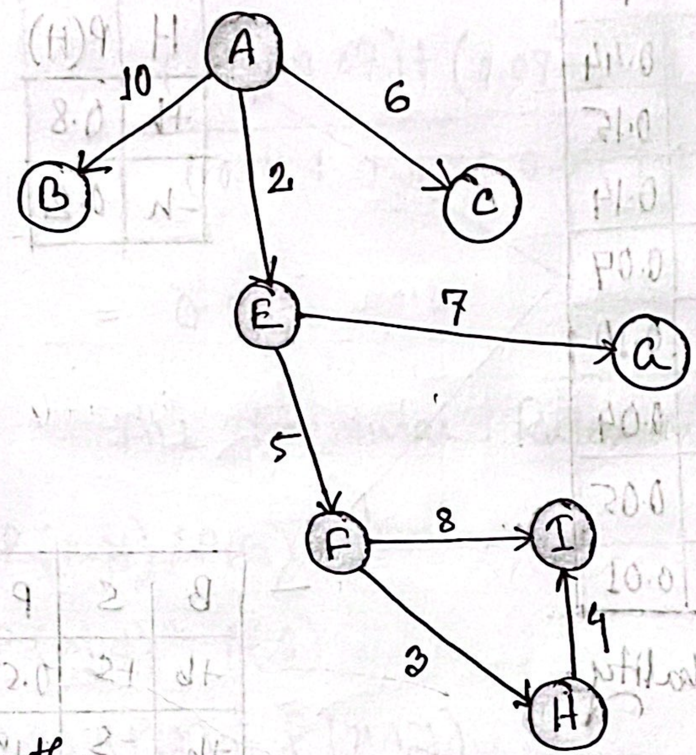
Total path cost = 15

Conclusion: Completeness but not optimal.

Subject: Date:

Technique 2:

Uniform Cost Search Algorithm



The actual path:

$A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{3} H \xrightarrow{4} I$

Total path cost: 14

UCS path:

$A \xrightarrow{2} E \xrightarrow{5} F \xrightarrow{3} H \xrightarrow{4} I$

Total path cost = 14

Conclusion: Completeness and optimal simultaneously.

Not always complete and optimal.

Subject :   Date :

Probability

H	B	S	P
+h	+b	+s	0.44
+h	+b	-s	0.15
+h	-b	+s	0.14
+h	-b	-s	0.07
-h	+b	+s	0.10
-h	+b	-s	0.04
-h	-b	+s	0.05
-h	-b	-s	0.01

H	P(H)
+h	0.8
-h	0.2

B	S	P
+b	+s	0.54
+b	-s	0.19
-b	+s	0.19
-b	-s	0.08

Conditional Probability

	+b		-b	
	+s	-s	+s	-s
+h	0.44	0.15	0.14	0.07
-h	0.10	0.04	0.05	0.01

(a) $P(+h \mid +b) = \dfrac{P(+h, +b)}{P(+b)} = \dfrac{0.44 + 0.15}{(0.44 + 0.15 + 0.1 + 0.04)} = 0.81$

(b) $P(-h, -s \mid +b) = \dfrac{P(-h, -s, +b)}{P(+b)} = \dfrac{0.04}{(0.44 + 0.15 + 0.1 + 0.04)} = 0.05$

(c) $P(-h \mid +s, -b) = \dfrac{P(-h, +s, -b)}{P(+s, -b)} = \dfrac{0.05}{(0.44 + 0.05)} = 0.26$

Subject :

Date :

(d) $$P(+h \vee -s \mid -b) = \frac{\left(P(+h,-b) + P(-s,-b) - P(+h,-s,-b)\right)}{P(-b)}$$

$$= \frac{(0.14+0.07)+(0.07+0.01)-0.07}{(0.14+0.05+0.07+0.01)}$$

$$= 0.81$$

ALL Formulae, Rules and Principles

$P(x|y) = P(x,y) / P(y)$

$P(x,y) = P(x|y) * P(y)$

$P(x \vee y) = P(x) + P(y) - P(x \cap y)$

$P(x,y) = P(x) * P(y)$

$P(x,y|z) = P(x|z) * P(y|z)$

$P(A') = 1 - P(A)$

$P(A|B) = P(A \cap B) / P(B)$

$P(A \cap B) = P(A) * P(B)$ if A and B are independent

$$P(A|B) = \frac{\left(P(B|A) * P(A)\right)}{P(B)}$$ (Bayes' Theorem)