Department of Information Technology

RJITC04

ARTIFICIAL INTELLIGENCE

DAY 4/ AI Algorithm Part 3

Content Overview

- Introduction to Problem Solving
- Beam Search Algorithm
- Simulated Annealing Algorithm
- Travelling Salesman Problem

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Problem Solving

- ***Problem solving*** is a process of generating solutions from observed data.

- a *‘problem’* is characterized by a set of *goals*,
- a set of *objects*, and
- a set of *operations*.

NAAC Re-Accredited 'A' Grade
विद्या अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

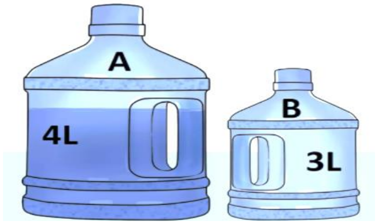
Problem space

- A ‘***problem space***’ is an abstract space.
- A problem space encompasses all *valid states* that can be generated by the application of any combination of *operators* on any combination of *objects*.
- The problem space may contain one or more *solutions*.
- A solution is combination of *operations* and *objects* that achieve the *goals*.

NAAC Re-Accredited 'A' Grade
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Water Jug Problem

- You are given two jugs - a 4-gallon one and a 3-gallon one,
- A pump which has unlimited water which you can use to fill the jug
- Ground on which water may be poured.
- Neither jug has any measuring markings on it.
- We can pour water from one jug to another.
- How can you get exactly 2 gallons of water in the 4-gallon jug?



NAAC Re-Accredited 'A' Grade
ESTD. 1963
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

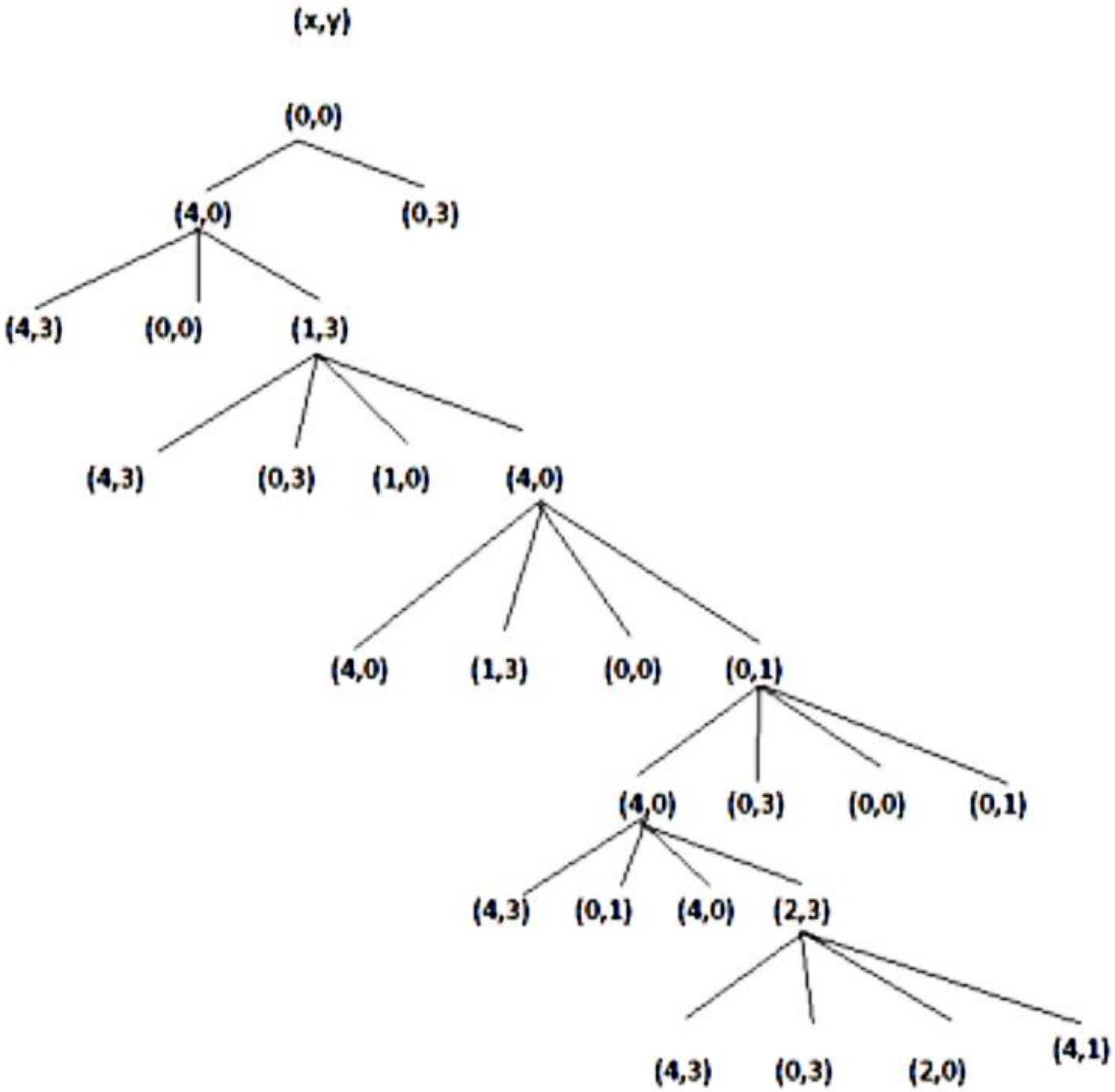
State Space Representation

- The state space for this problem can be described as the set of ordered pairs of integer (x,y).
- x- water in 4- gallon jug
- y- water in 3- gallon jug
- The Start state is (0,0).
- The Goal state is (2,y)

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

S.No	Initial State	Rules/Conditions	Final state	Description of action taken
1.	**(x,y)**	**If x < 4**	**(4,y)**	**Fill the 4 gallon jug completely**
2.	**(x,y)**	**if y < 3**	**(x,3)**	**Fill the 3 gallon jug completely**
3.	**(x,y)**	**If x > 0**	**(x-d,y)**	**Pour some part from the 4 gallon jug**
4.	**(x,y)**	**If y > 0**	**(x,y-d)**	**Pour some part from the 3 gallon jug**
5.	**(x,y)**	**If x > 0**	**(0,y)**	**Empty the 4 gallon jug**
6.	**(x,y)**	**If y > 0**	**(x,0)**	**Empty the 3 gallon jug**
7.	**(x,y)**	**If (x+y) < 7**	**(4, y-[4-x])**	**Pour some water from the 3 gallon jug to fill the 4 gallon jug**
8.	**(x,y)**	**If (x+y) < 7**	**(x-[3-y],y)**	**Pour some water from the 4 gallon jug to fill the 3 gallon jug.**
9.	**(x,y)**	**If (x+y) < 4**	**(x+y,0)**	**Pour all water from 3 gallon jug to the 4 gallon jug**
10.	**(x,y)**	**if (x+y) < 3**	**(0, x+y)**	**Pour all water from the 4 gallon jug to the 3 gallon jug**

Gallons in the 4-gallon jug	Gallons in the 3-gallon jug	Rule applied
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5 or 12
0	2	9 or 11
2	0	--



Water Jug problem.py - D:/python/Water Jug problem.py (3.9.2)

File Edit Format Run Options Window Help

```python
print("Water Jug Problem")
x=int(input("Enter X:"))
y=int(input("Enter Y:"))
while True:
    rno=int(input("Enter the Rule No"))
    if rno==1:
        if x<4:x=4
    if rno==2:
        if y<3:y=3
    if rno==3:
        if x>0:x=0
    if rno==4:
        if y>0:y=0
    if rno==5:
        if x+y>= 4 and y>0:x,y=4,y-(4-x)
    if rno==6:
        if x+y>=3 and x>0:x,y=x-(3-y),3
    if rno==7:
        if x+y<=4 and y>0:x,y=x+y,0
    if rno==8:
        if x+y<=3 and x>0:x,y=0,x+y
    print("X =" ,x)
    print("Y =" ,y)
    if (x==2):
        print(" The result is a Goal state")
        break
```

```
==================== RESTART: D:/p
Water Jug Problem
Enter X:0
Enter Y:0
Enter the Rule No2
X = 0
Y = 3
Enter the Rule No7
X = 3
Y = 0
Enter the Rule No2
X = 3
Y = 3
Enter the Rule No5
X = 4
Y = 2
Enter the Rule No3
X = 0
Y = 2
Enter the Rule No7
X = 2
Y = 0
 The result is a Goal state
>>>
```

```
================================
Water Jug Problem
Enter X:0
Enter Y:0
Enter the Rule No1
X = 4
Y = 0
Enter the Rule No6
X = 1
Y = 3
Enter the Rule No4
X = 1
Y = 0
Enter the Rule No8
X = 0
Y = 1
Enter the Rule No1
X = 4
Y = 1
Enter the Rule No6
X = 2
Y = 3
 The result is a Goal state
```

Search

- A '***search***' refers to the search for a solution in a problem space.
- The problem can then be solved by using the *rules*, in combination with an appropriate *control strategy*, to move through the *problem space* until a *path* from an *initial state* to a *goal state* is found. This process is known as ***'search'***.
- *Search* is a general mechanism that can be used when a more direct method is not known.

NAAC Re-Accredited 'A' Grade
ESTD. 1962
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

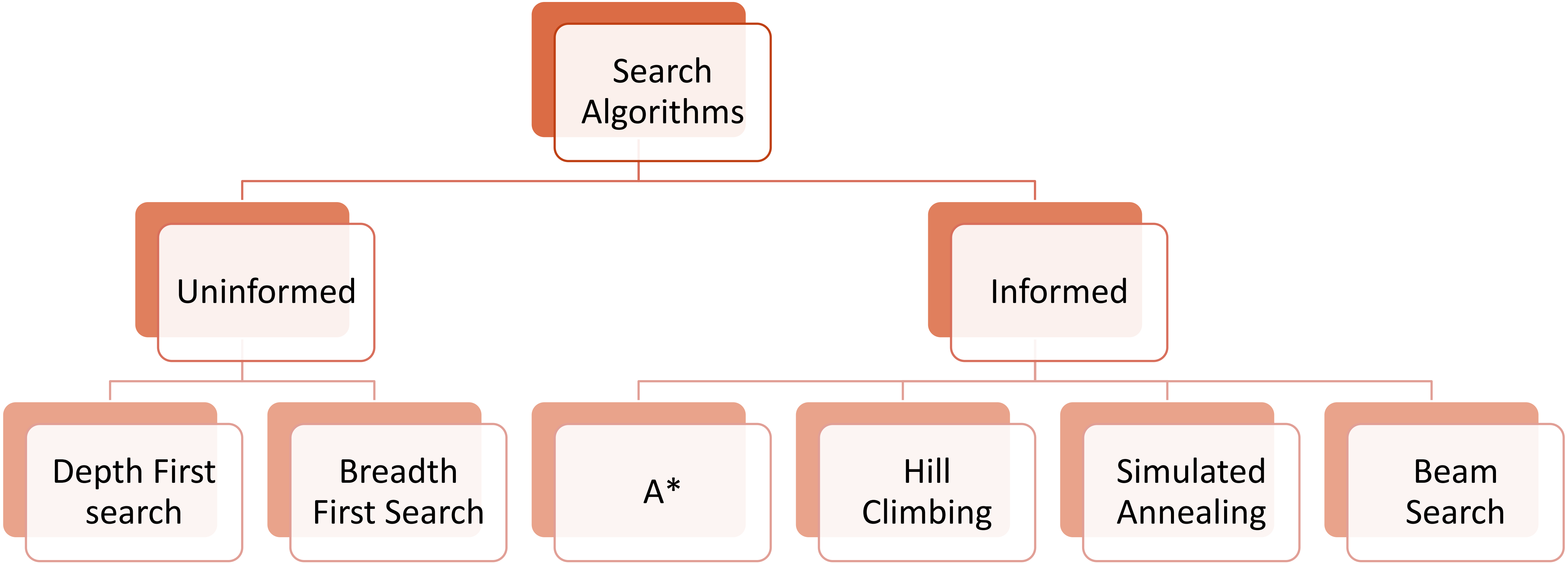
Search Algorithms

Uninformed search algorithms or blind search, exhaustive or brute-force search,

- Searches through the search space across all possible candidates for the solution checking whether each candidate satisfies the problem's statement.
- It uses no information about the problem to guide the search

Informed search algorithms use heuristic functions that are specific to the problem,

- It applies heuristic function to guide the search through the search space
- Guesses the distance to a goal state and it reduces the amount of time spent in searching.
- Heuristic algorithms are not really intelligent; they appear to be intelligent because they achieve better performance.


Search Algorithms
Uninformed
Informed
Depth First search
Breadth First Search
A*
Hill Climbing
Simulated Annealing
Beam Search

Beam Search Algorithm

- Search Algorithms like BFS, DFS and A* etc. are infeasible on large search spaces.
- Beam Search was developed in an attempt to achieve the optimal (or sub-optimal) solution.
- It is a heuristic approach where only the most promising β nodes (instead of all nodes) at each step of the search are retained for further branching.
- β is called Beam Width.
- Beam search is an optimization of best-first search that reduces its memory requirements.
- It is used in many machine translation systems.

NAAC Re-Accredited 'A' Grade
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

- The idea is that you just keep around those states that are relatively good, and just forget the rest.
- **Local beam search: somewhat similar to Hill Climbing:**
  - Start from **N** initial states.
  - Expand all **N** states and keep **k** best successors.
- **Local Beam Search Algorithm**
- Keep track of *k* states instead of one
  - Initially: *k* random states
  - Next: determine all successors of *k* states
    - Extend **all paths** one step
    - Reject all paths with loops
    - Sort all paths in queue by **estimated** distance to goal
  - If any of successors is goal → finished
  - Else select *k* best from successors and repeat.

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Hill Climbing Vs. Beam Search

- Hill climbing just explores all nodes in one branch until goal found or not being able to explore more nodes.
- Beam search explores more than one path together. A factor **k (β Beam Width)** is used to determine the number of branches explored at a time.
- If k=2, then two branches are explored at a time. For k=4, four branches are explored simultaneously.
- The branches selected are the **best branches** based on the used **heuristic evaluation function**.

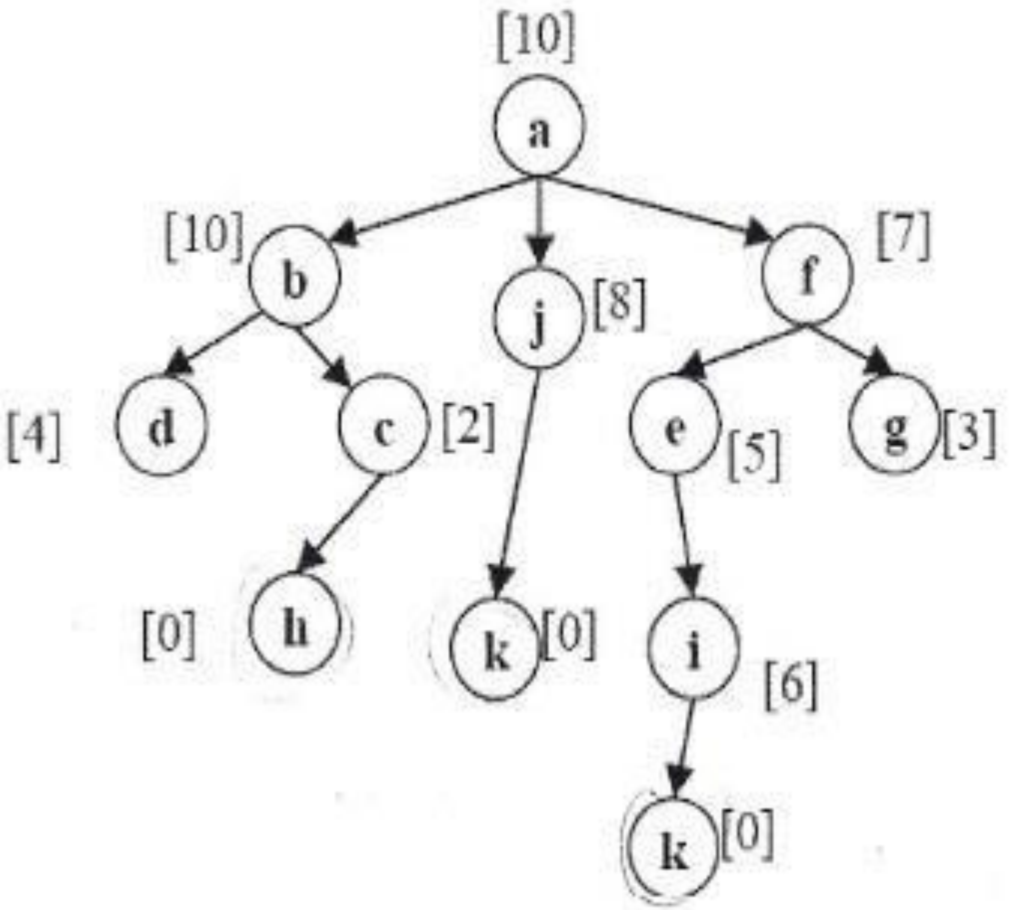
NAAC Re-Accredited 'A' Grade
विद्या अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Beam Search, k=2

Start Node –A

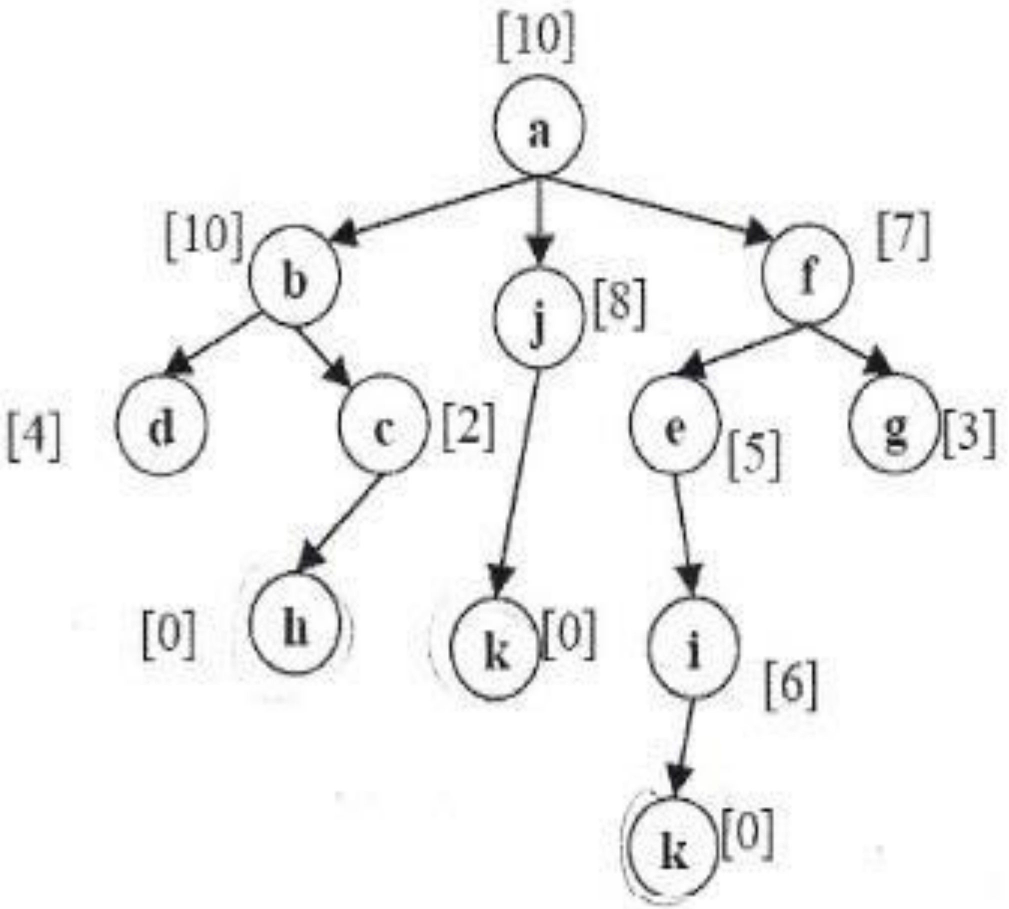
Goal Node – K

Current	Children
a	---



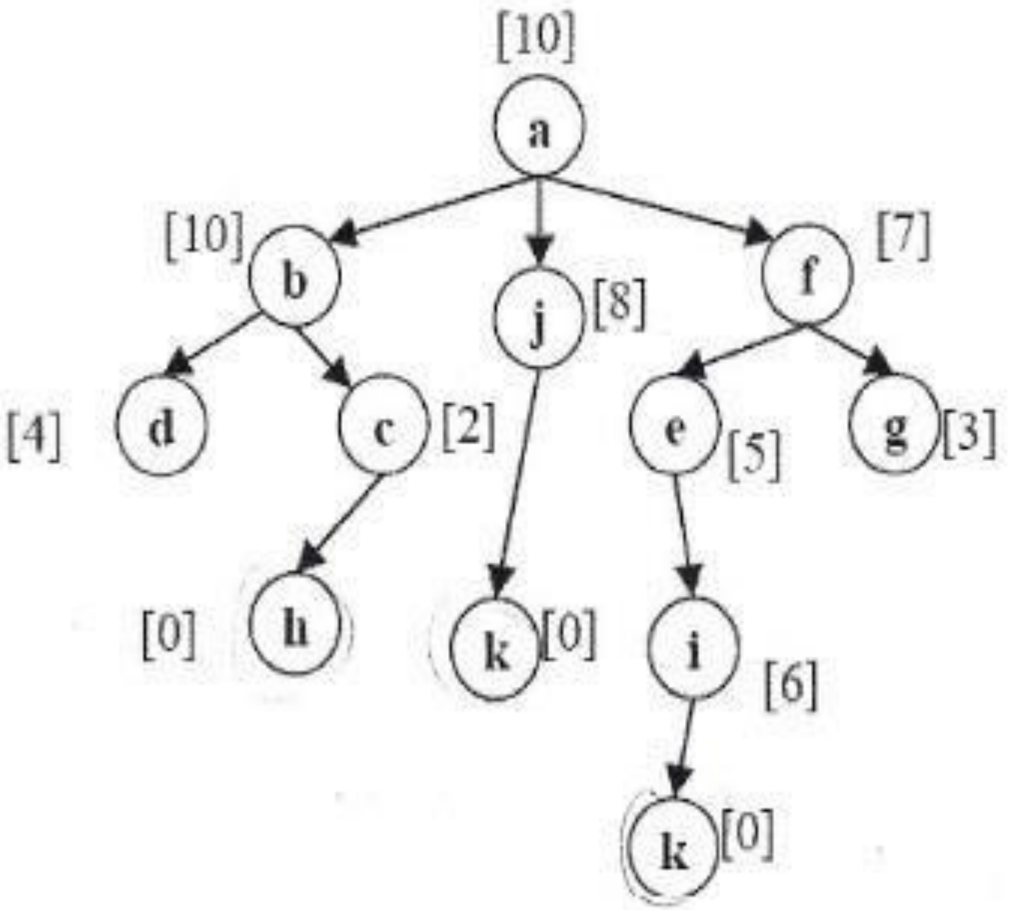
Beam Search
Goal Node - K

Current	Children
a	---



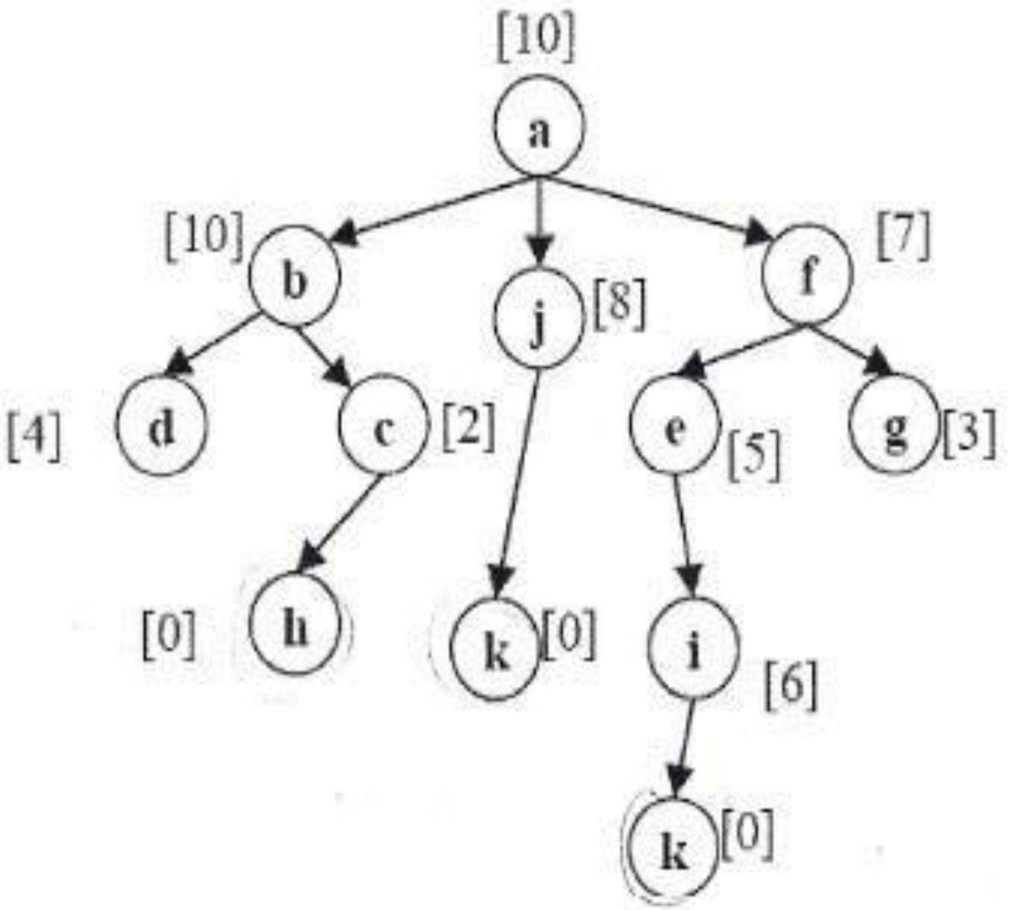
Beam Search
Goal Node - K

Current	Children
a	---
a	



Beam Search
Goal Node - K

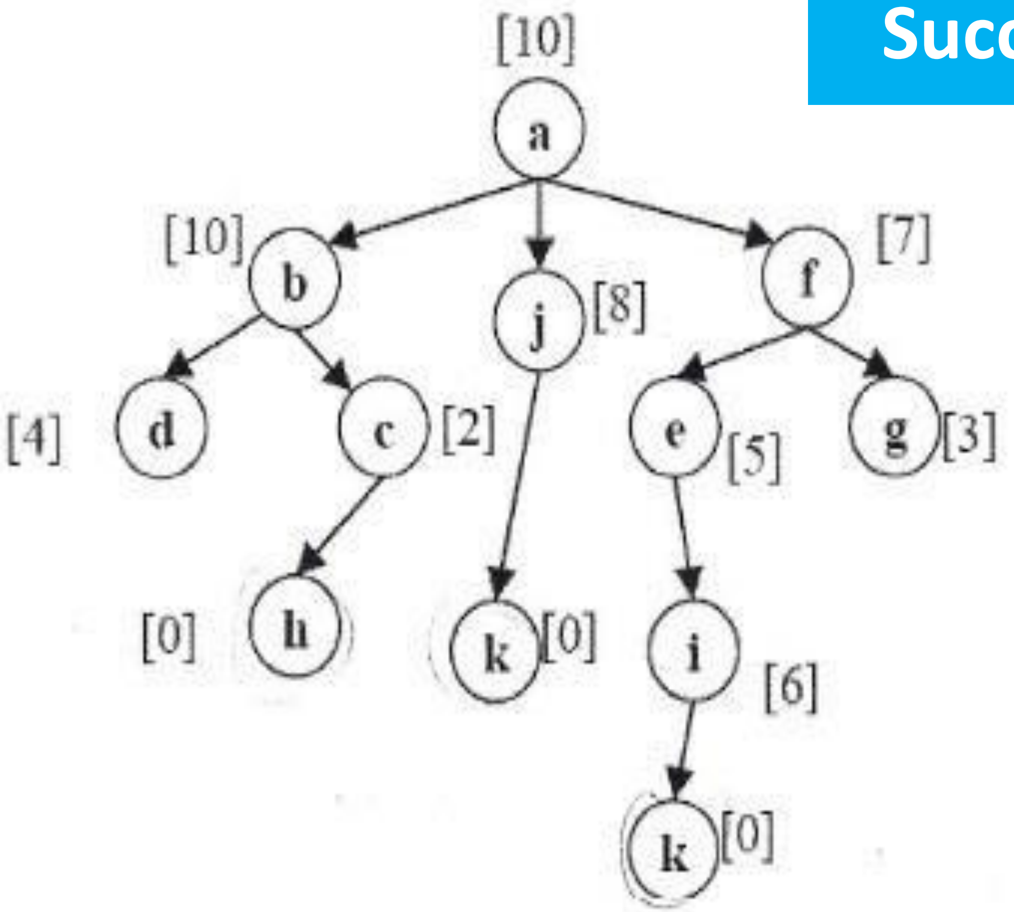
Current	Children
a	---
a	f_7, j_8, b_{10}



Beam Search
Goal Node - K

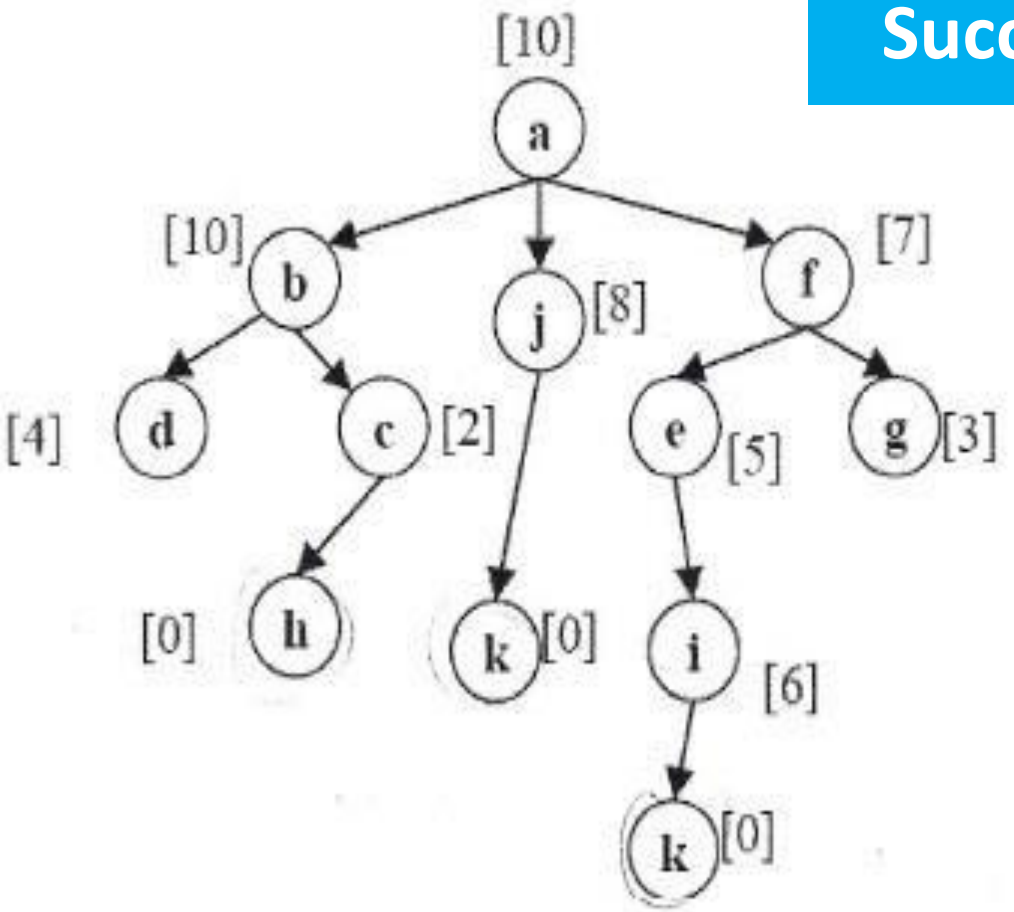
Best k Successors

Current	Children
a	---
a	f_7, j_8, b_{10}



ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

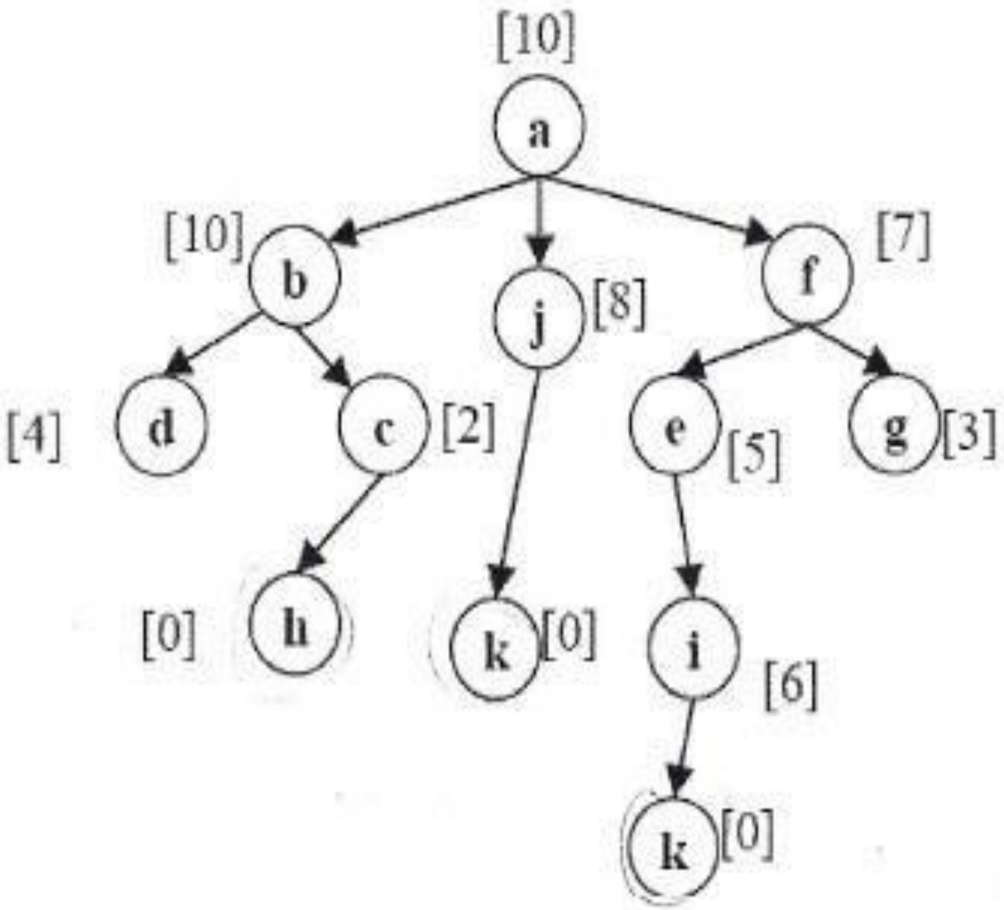
Beam Search
Goal Node - K



Best k Successors

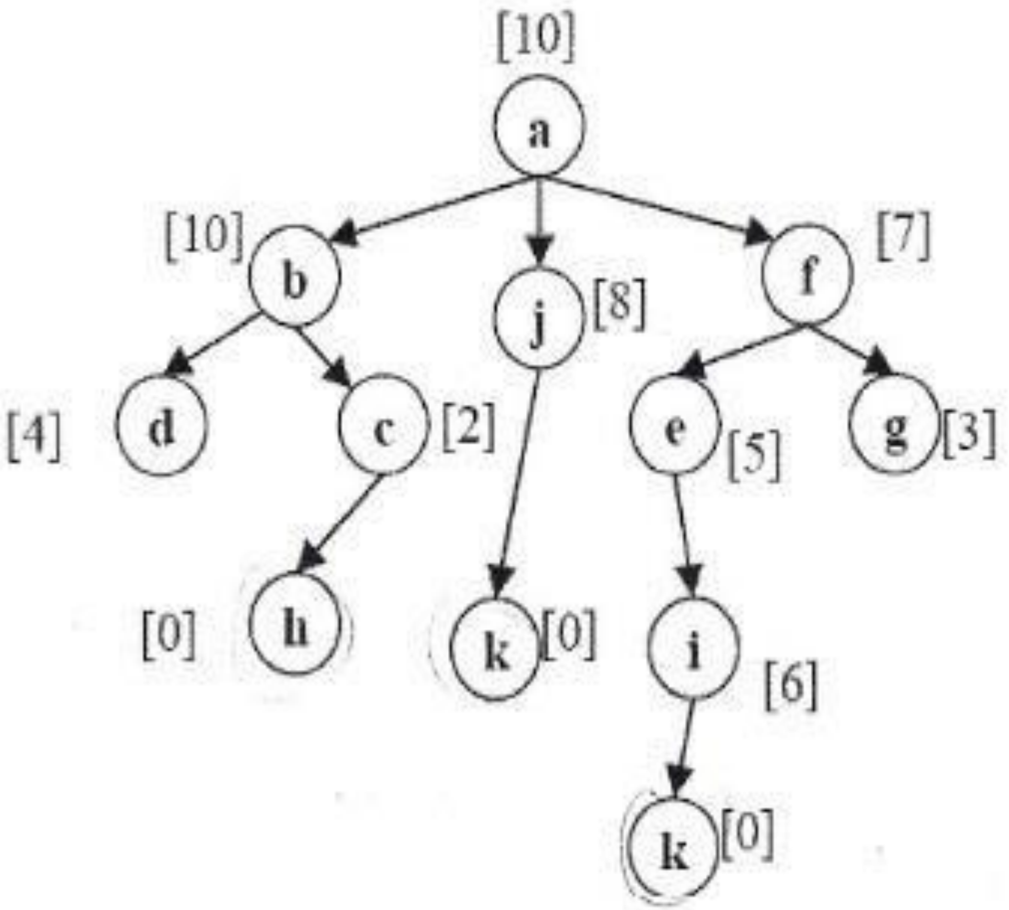
Current	Children
a	---
a	f_7, j_8, b_{10}

Beam Search
Goal Node - K



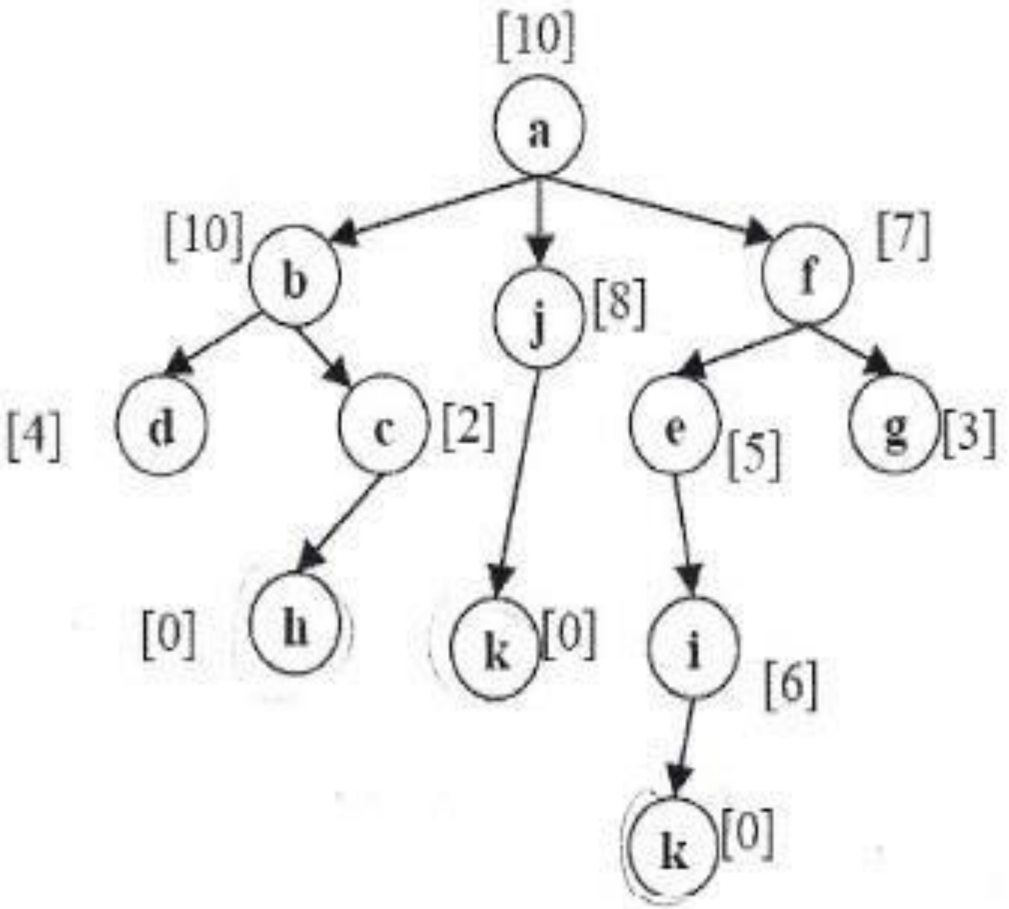
Current	Children
a	---
a	f_7, j_8, b_{10}
f j	

Beam Search
Goal Node - K



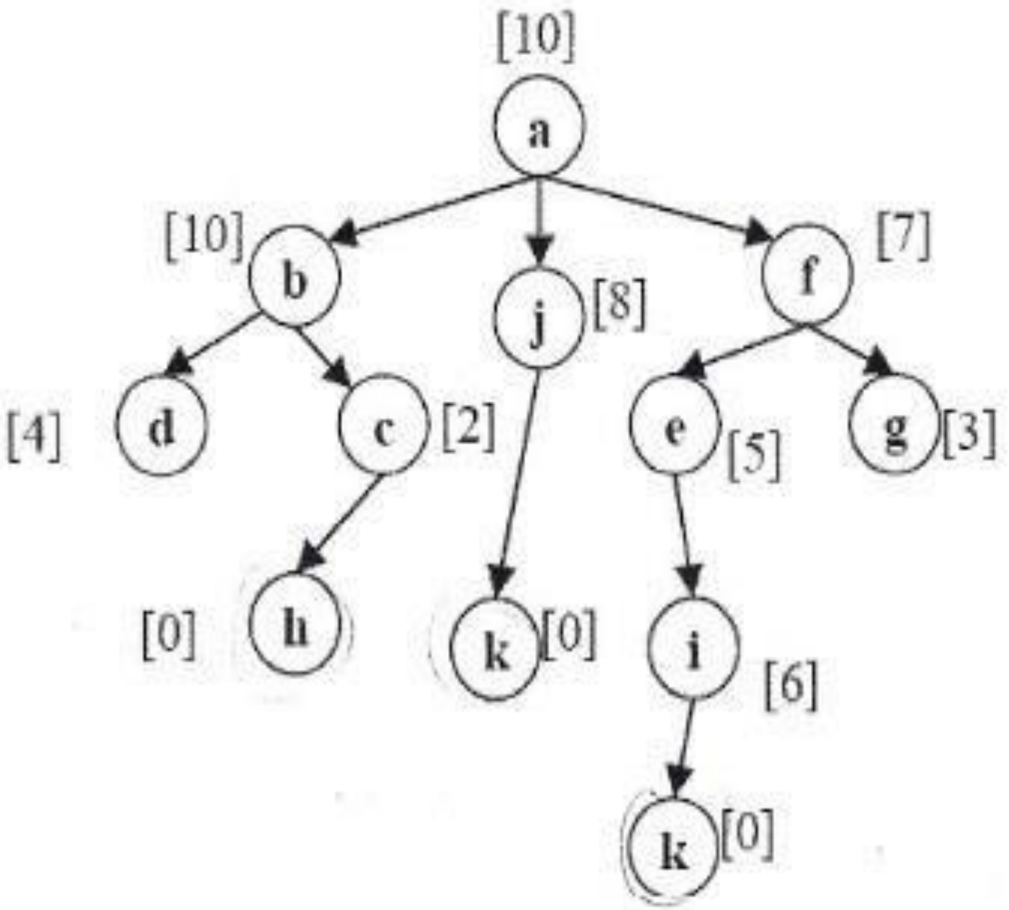
Current	Children
a	---
a	f_7, j_8, b_{10}
f j	

Beam Search
Goal Node - K



Current	Children
a	---
a	f_7, j_8, b_{10}
f j	

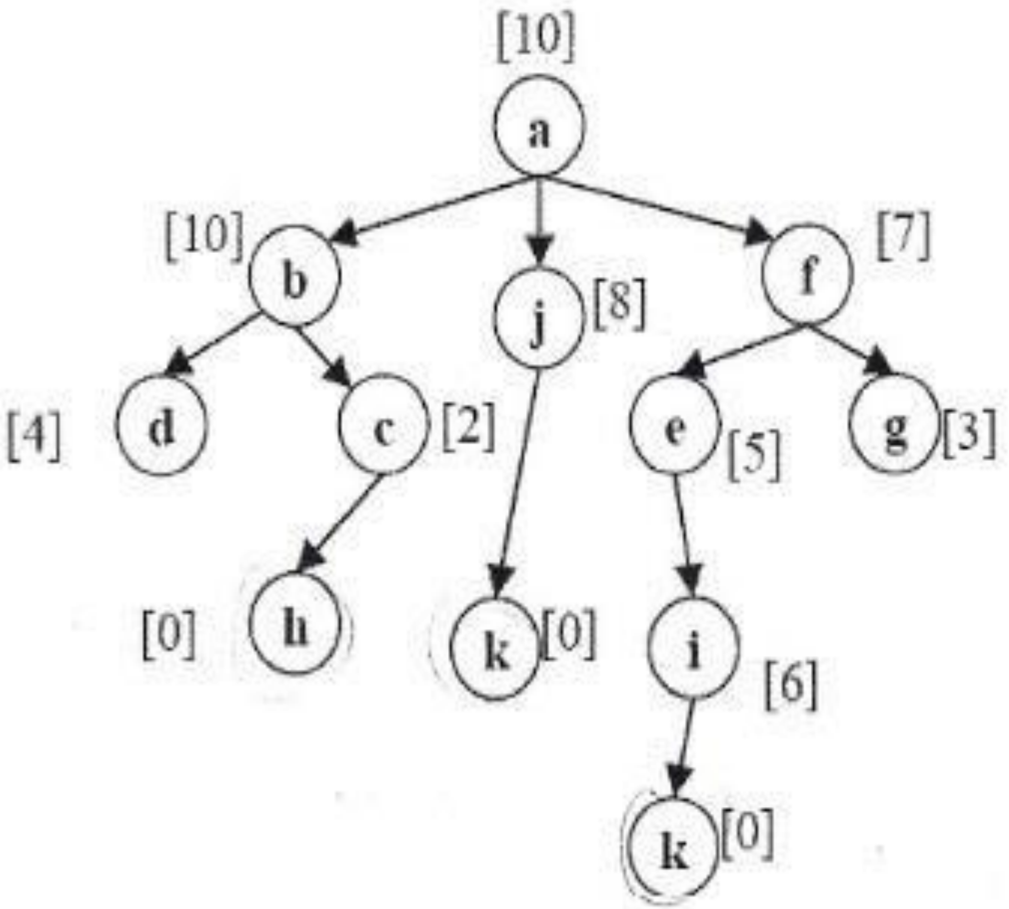
Beam Search
Goal Node - K



Current	Children
a	---
a	f_7, j_8, b_{10}
f	g_3, e_5
j	k_0

ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

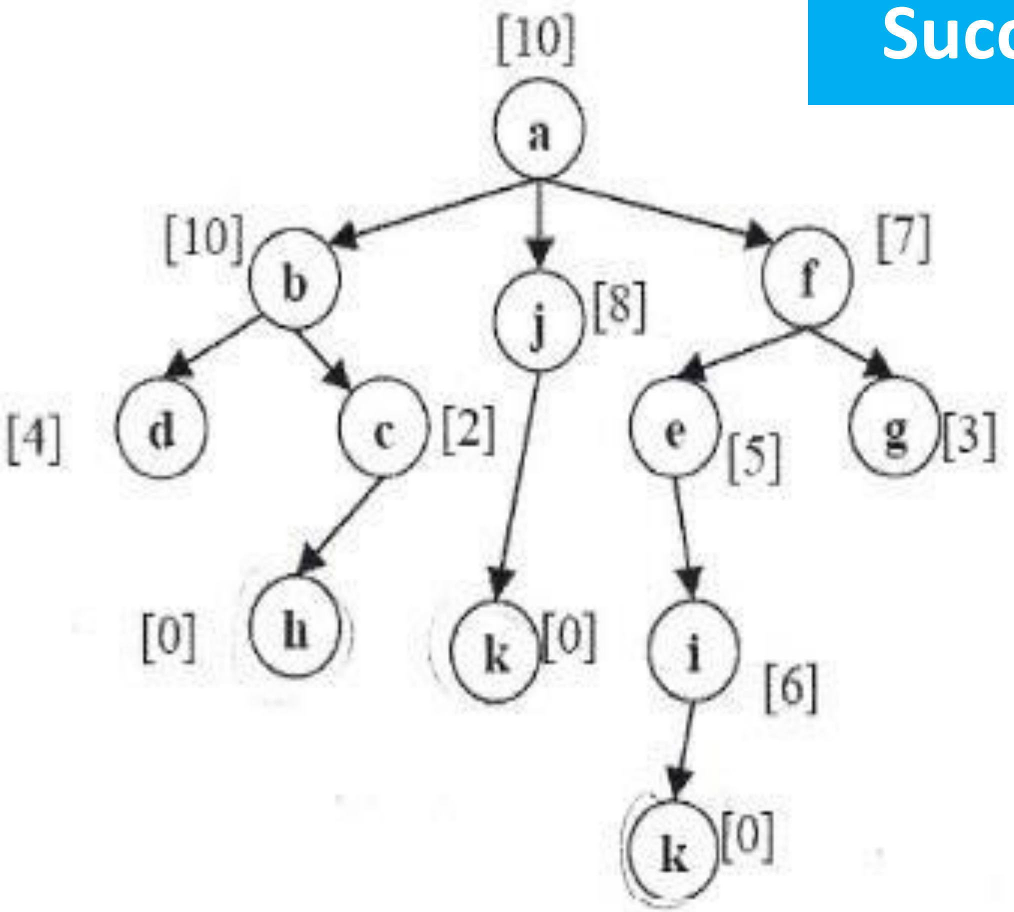
Beam Search
Goal Node - K



Current	Children
a	---
a	f_7, j_8, b_{10}
f	g_3, e_5
j	k_0

k_0, g_3, e_5

Beam Search
Goal Node - K



Best k Successors

Current	Children
a	---
a	f_7, j_8, b_{10}

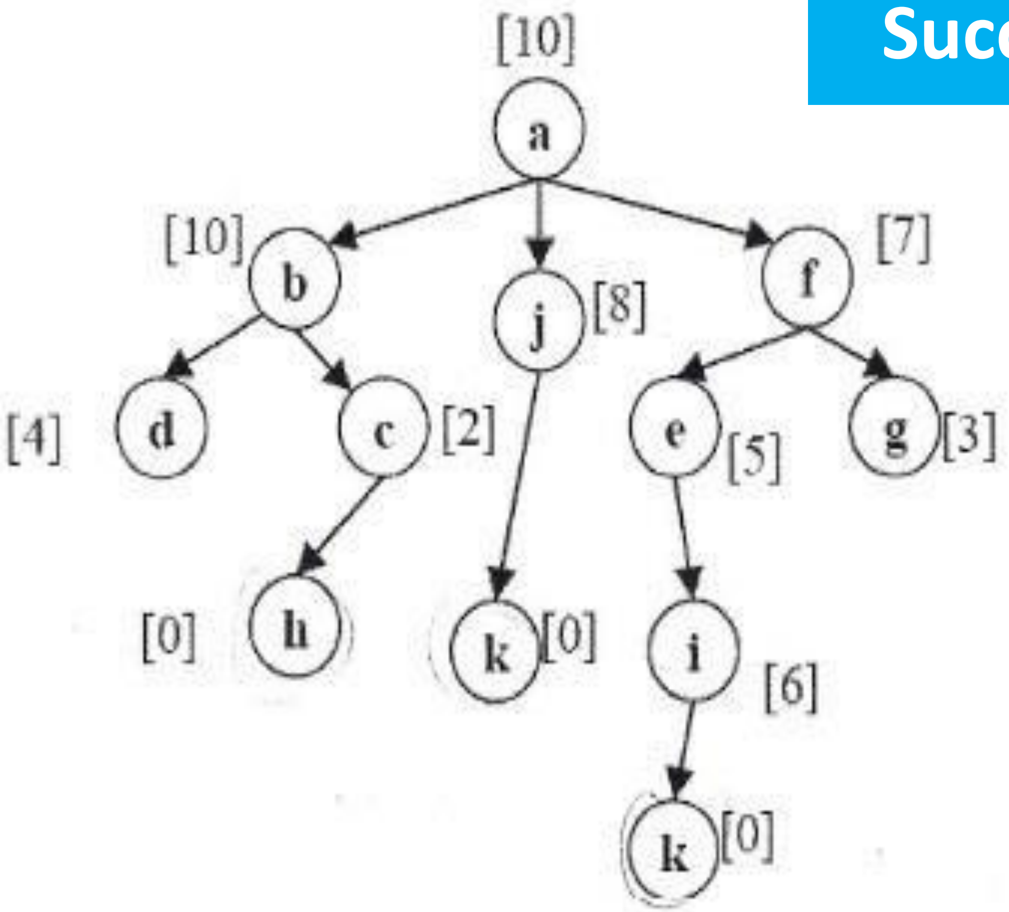
f | j

g_3, e_5 | k_0

k_0, g_3, e_5

ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

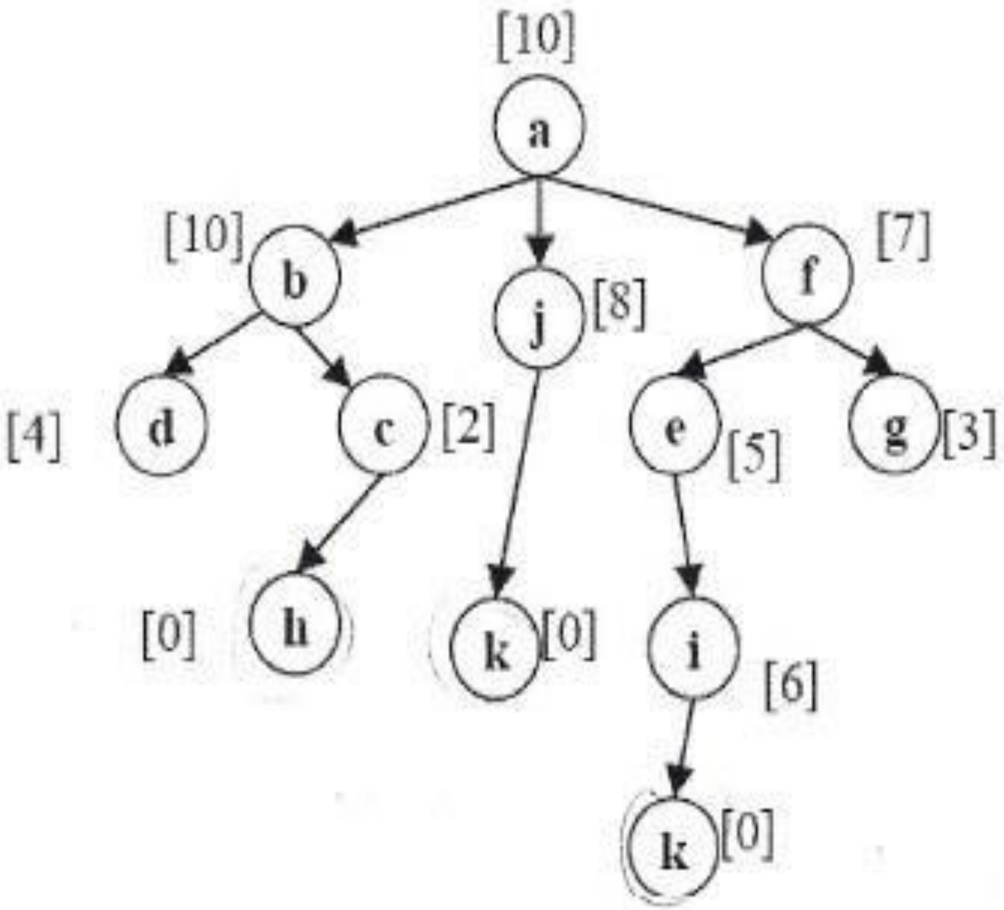
Beam Search
Goal Node - K



Best k Successors

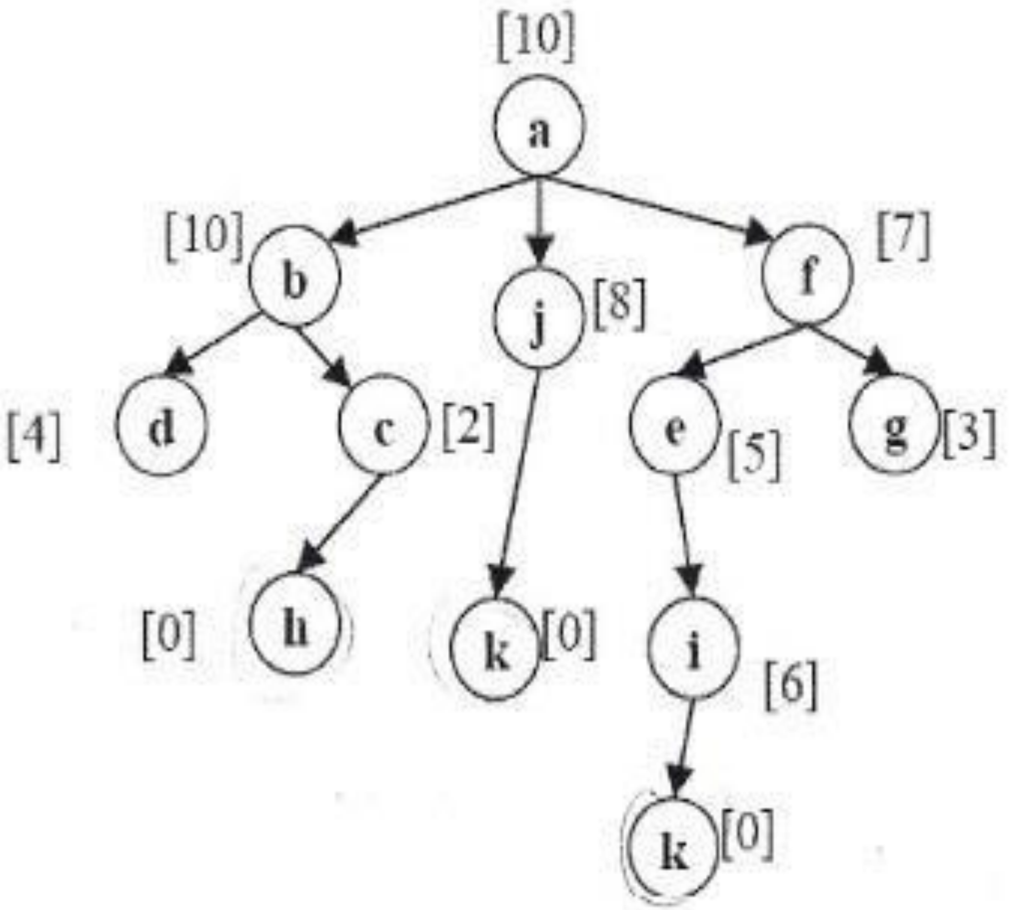
Current	Children
a	---
a	f_7, j_8, b_{10}
f	g_3, e_5
j	k_0
	k_0, g_3, e_5

Beam Search
Goal Node - K



Current	Children
a	---
a	f_7, j_8, b_{10}
f, j	
g_3, e_5	k_0
k_0, g_3, e_5	
k, g	

Beam Search
Goal Node - K



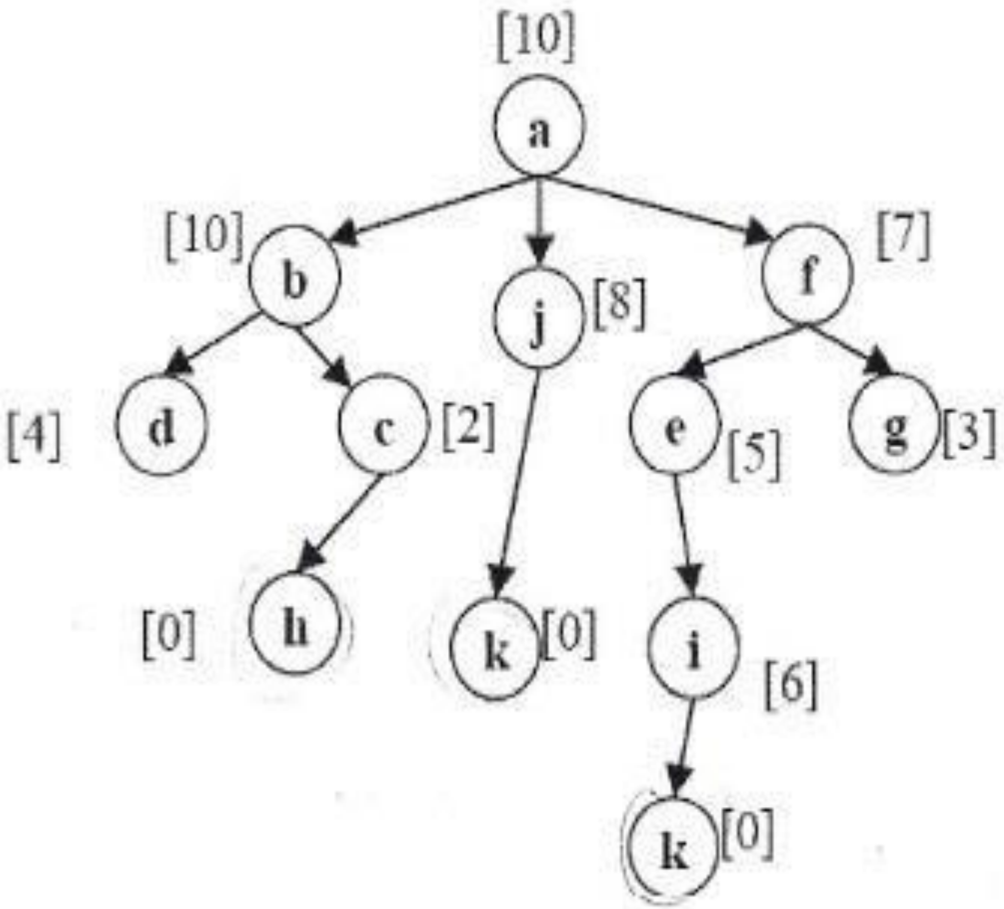
Current	Children
a	---
a	f_7, j_8, b_{10}
f	g_3, e_5
j	k_0

k_0, g_3, e_5

k g

ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

Beam Search
Goal Node - K



Current	Children
a	---
a	f_7, j_8, b_{10}

f j

g_3, e_5 k_0

k_0, g_3, e_5

k g

GOAL

Applications of Beam Search

- Job Scheduling - early/tardy scheduling problem
- Phrase-Based Translation Model
- Speech recognition, vision, planning, and machine learning

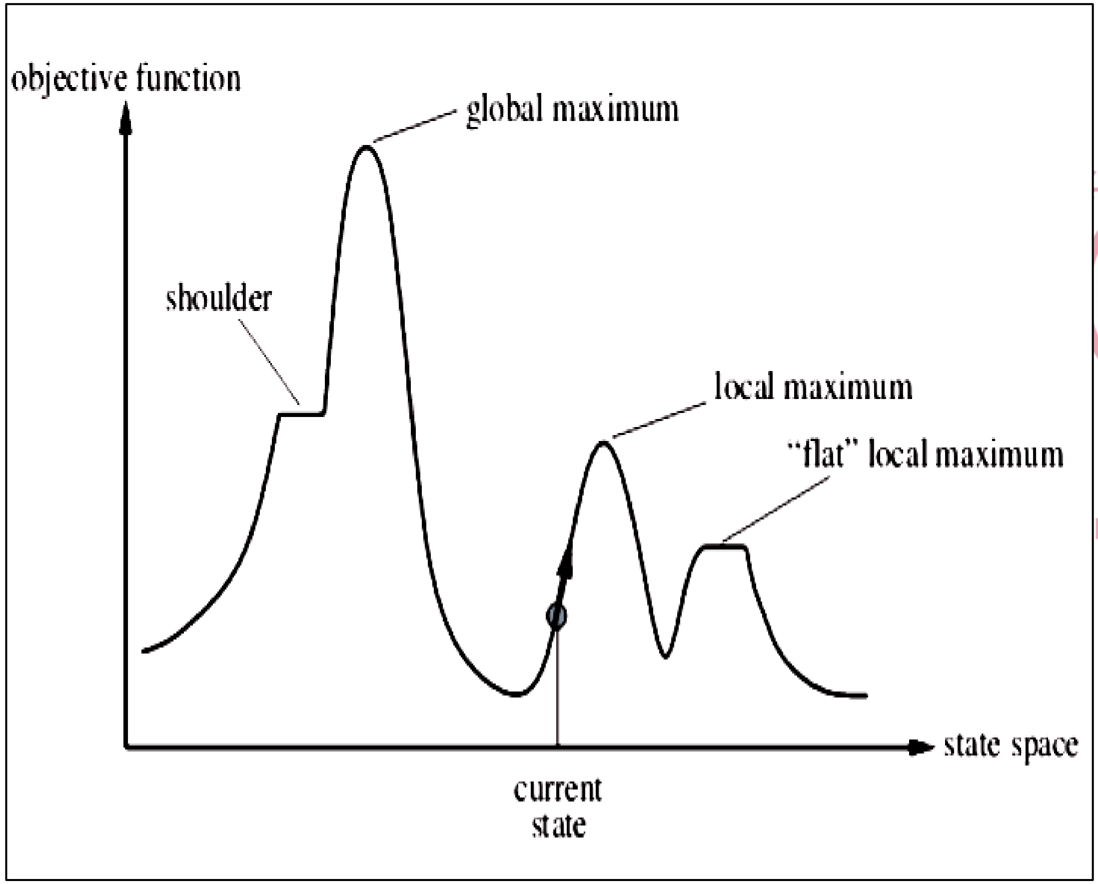
NAAC Re-Accredited 'A' Grade
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

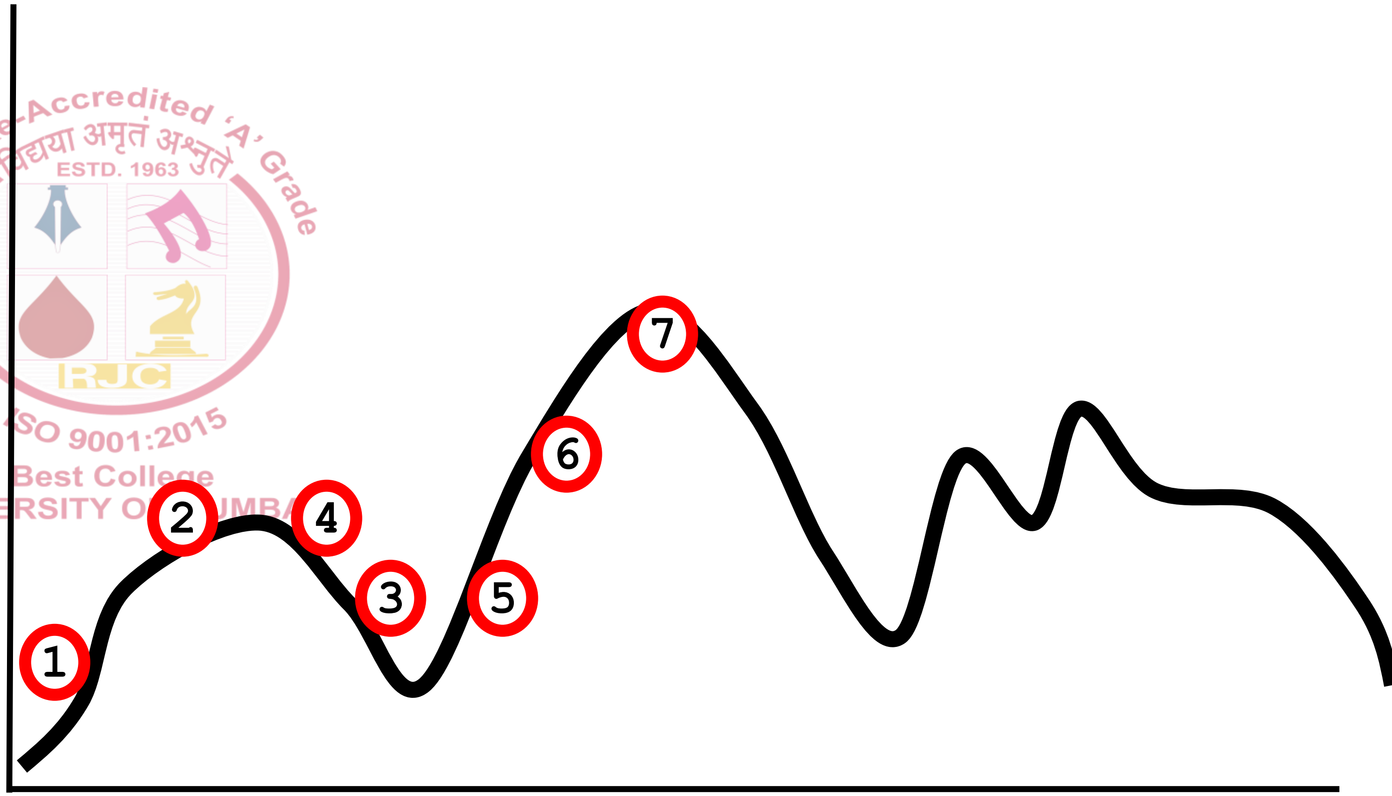
Simulated Annealing

- Motivated by the physical annealing process
- Annealing: harden metals and glass by heating them to a high temperature and then gradually cooling them
- At the start, make lots of moves and then gradually slow down
- More formally…
  - Instead of picking the best move (as in Hill Climbing),
  - Generate a random new neighbor from current state,
  - If it's better take it,
    - If it's worse then take it with some probability proportional to the temperature and the delta between the new and old states,
    - Probability gets smaller as time passes and by the amount of "badness" of the move,
- Compared to hill climbing the main difference is that SA allows downwards steps; (moves to higher cost successors).
- Simulated annealing also differs from hill climbing in that a move is selected at random and then decides whether to accept it.

NAAC Accredited 'A' Grade
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Simulated Annealing





Simulated Annealing

- The probability of making a downhill move decreases with time (length of the exploration path from a start state).
- The choice of probability distribution for allowing downhill moves is derived from the physical process of annealing metals (cooling molten metal to solid minimal- energy state).
- During the annealing process in metals, there is a probability p that a transition to a higher energy state (which is sub-optimal) occurs. Higher energy implies lower value. The probability of going to a higher energy state is $e^{\Delta/T}$ where
  - Δ = (energy of next state) – (energy of current state)
  - T is the temperature of the metal.
- p is higher when T is higher, and movement to higher energy states become less likely as the temperature cools down.
- For both real and simulated annealing, the rate at which a system is cooled is called the *annealing schedule*, or just the "schedule."

NAAC Re-Accredited 'A' Grade
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Simulated Annealing Algorithm

- Select a start node (root node).
- Randomly select a child of the current node, calculate a value reflecting how good such child is like $\boldsymbol{value}(\boldsymbol{node}) = -\boldsymbol{heuristic}(\boldsymbol{node})$.
- Select the child if it is better than the current node. Else try another child.

  A node is better than the current node if $\boldsymbol{\Delta E} = \boldsymbol{value}(\boldsymbol{next}) - \boldsymbol{value}(\boldsymbol{current}) > \mathbf{0}$.

  Else if $\boldsymbol{\Delta E} < \mathbf{0}$, then try to find another child.

- If the child was not better than the current node then it will be selected with probability equal to $\mathbf{p} = \boldsymbol{e}^{\frac{\boldsymbol{\Delta E}}{\boldsymbol{T}}}$

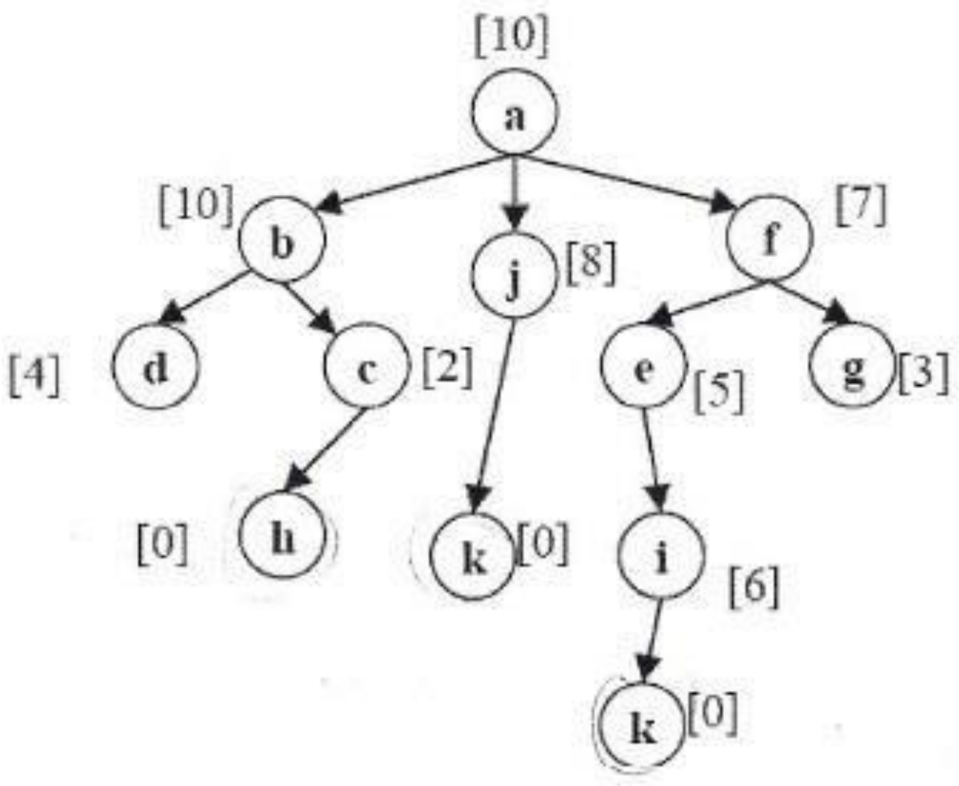
  where $\boldsymbol{\Delta E} = \boldsymbol{value}[\boldsymbol{next}] - \boldsymbol{value}[\boldsymbol{current}]$

  $\boldsymbol{T}$ is a temperature.

- Stop if no improvement can be found or after a fixed time.

Simulated Annealing Example – T=10

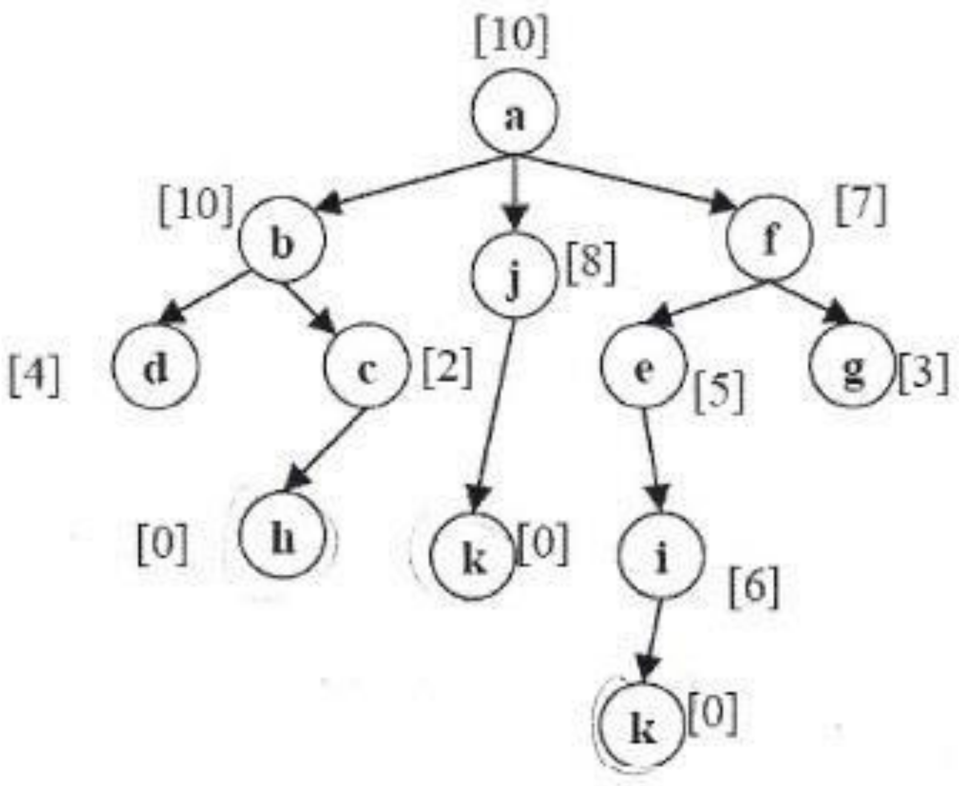
Current	Children
a	



Simulated Annealing Starting from Node a

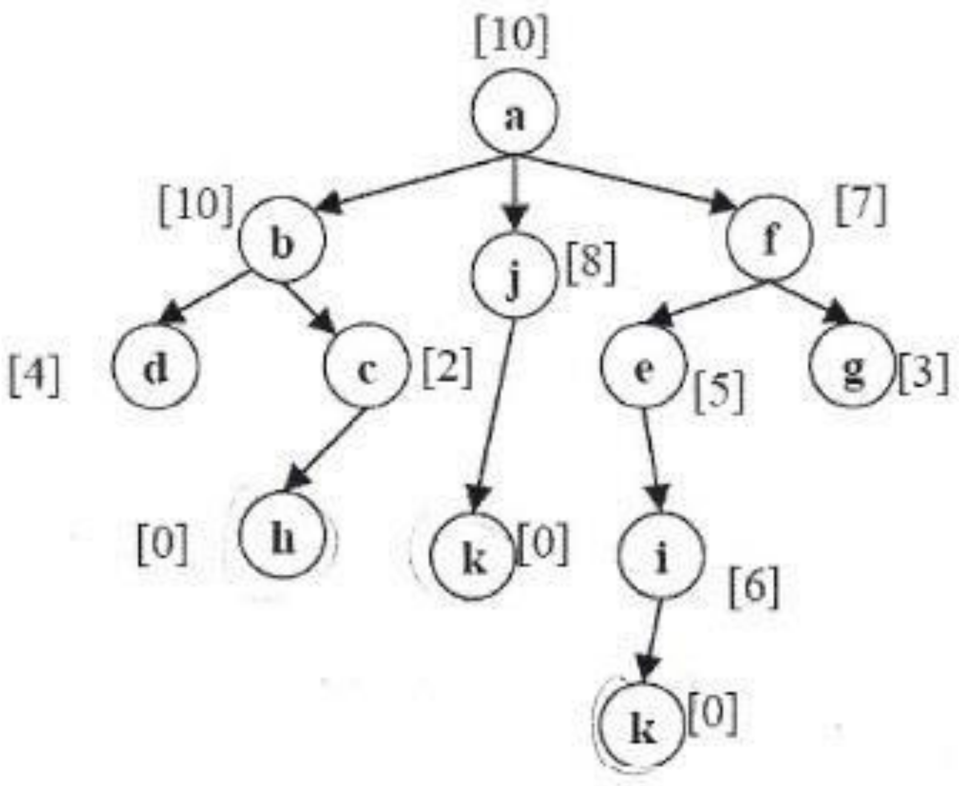
Current | Children

a



Simulated Annealing Starting from Node a

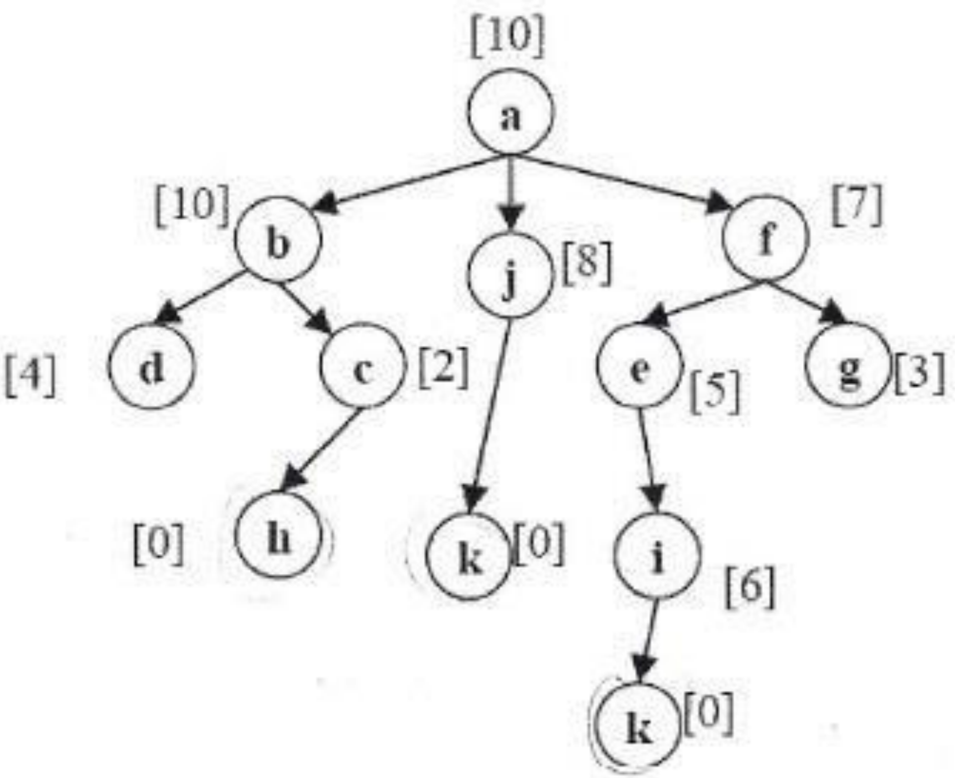
Current	Children
a	



ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

Simulated Annealing

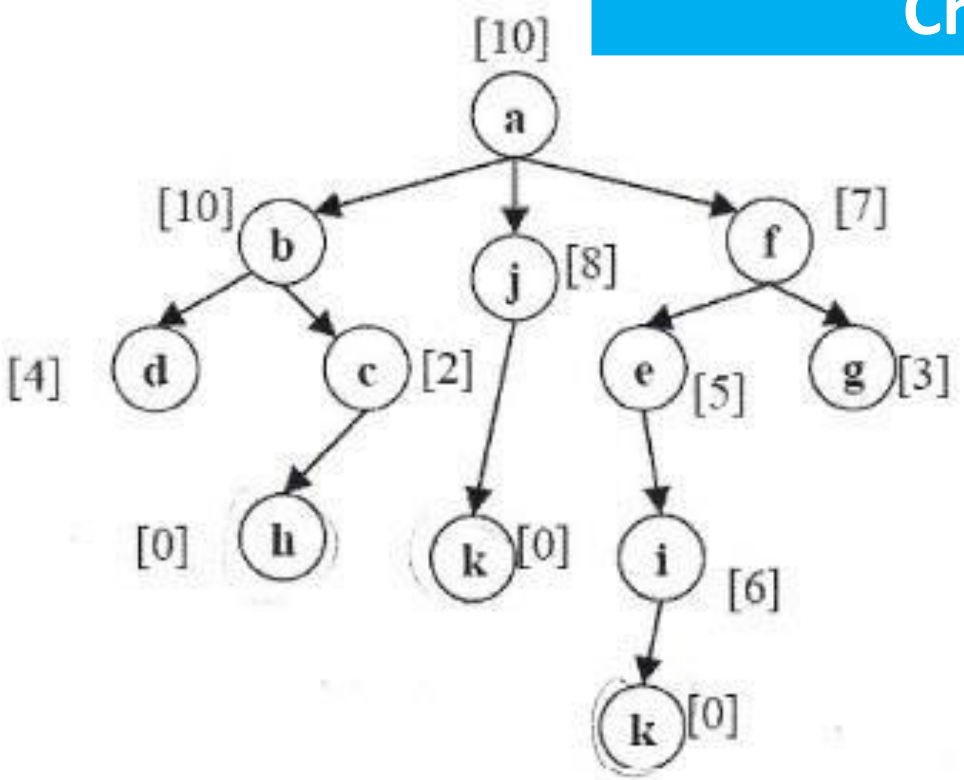
Current	Children
a	---
a	f_7, j_8, b_{10}



Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}

Randomly Select a Child

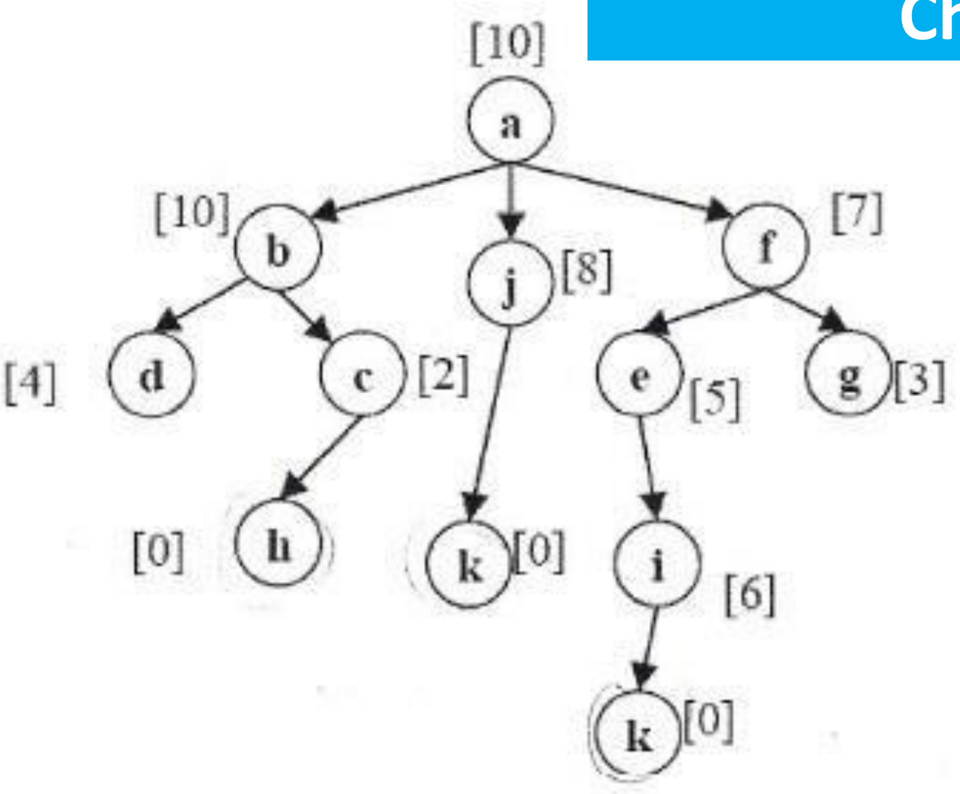


ESTD. 1963
9001:2015
st College
ITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7 , j_8, b_{10}

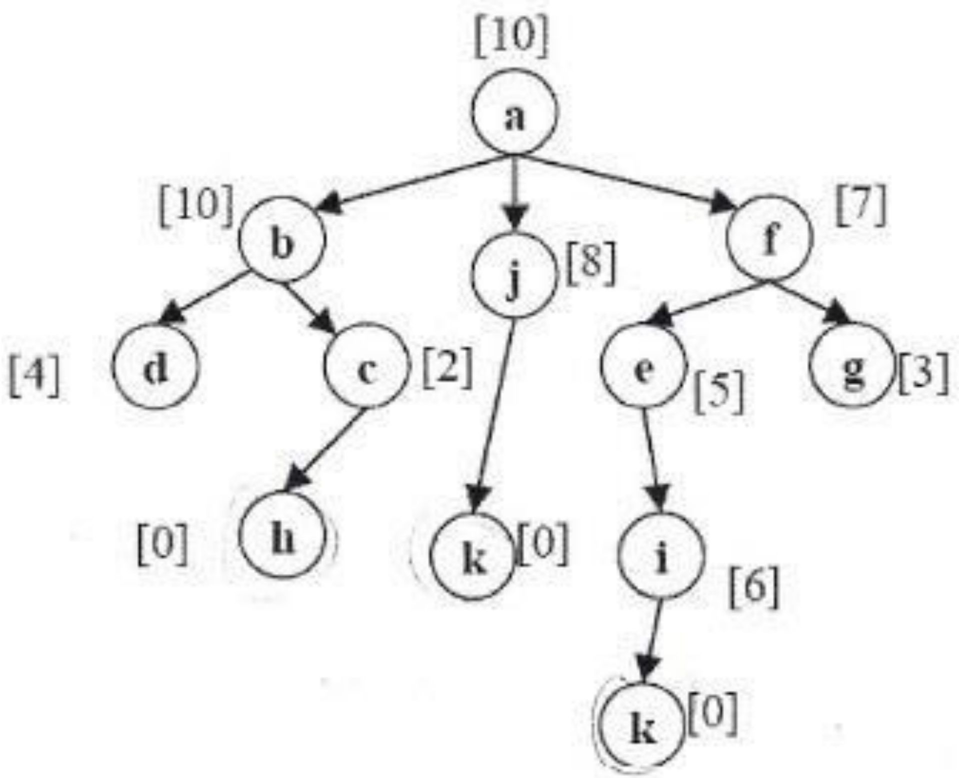
Randomly Select a Child



Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}

Check if next node f_7 is better than current node

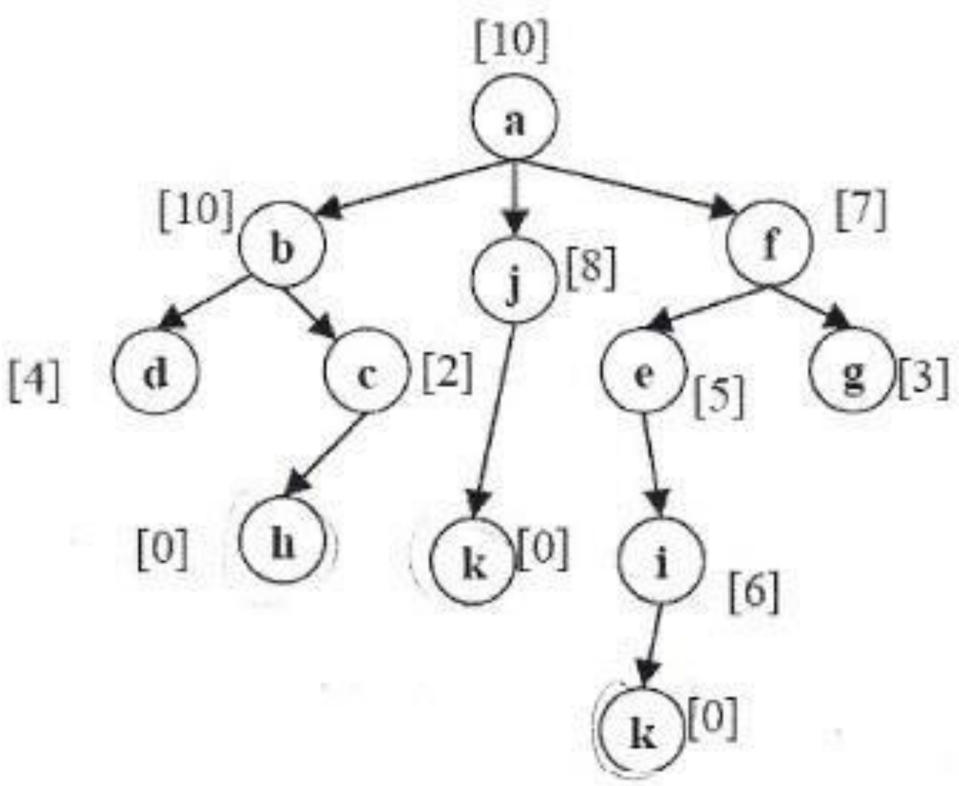


Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}

Check if next node f_7 is better than current node

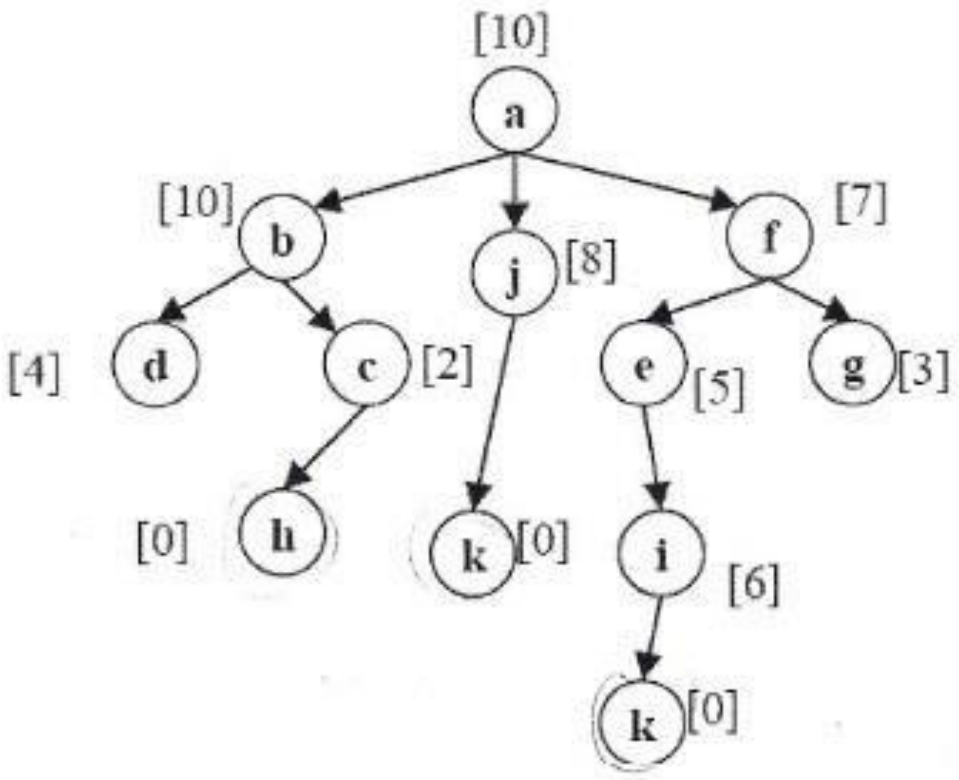
$\Delta E > 0$



ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}

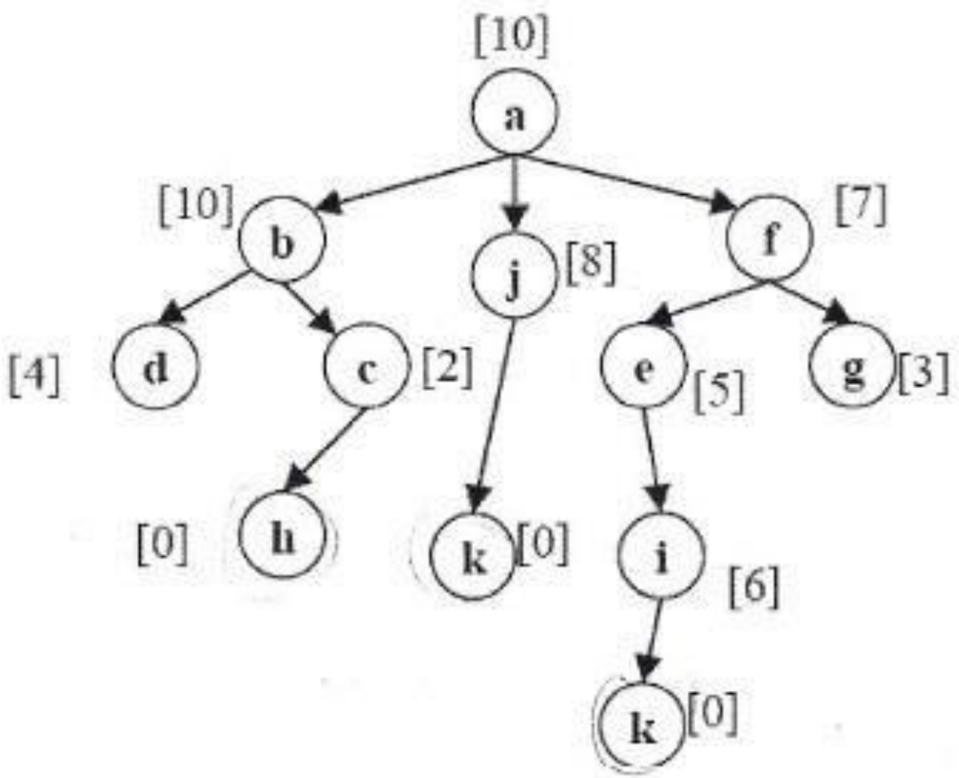


Check if next node f_7 is better than current node

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}



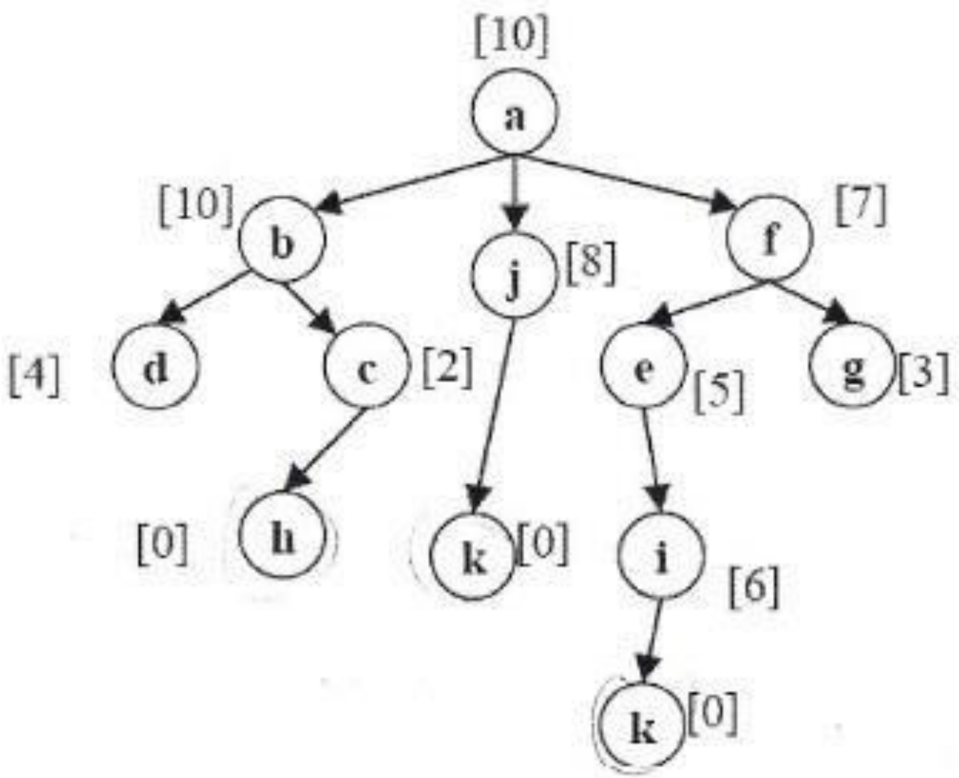
Check if next node f_7 is better than current node

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(f_7) - \text{value}(a_{10})$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}



Check if next node f_7 is better than current node

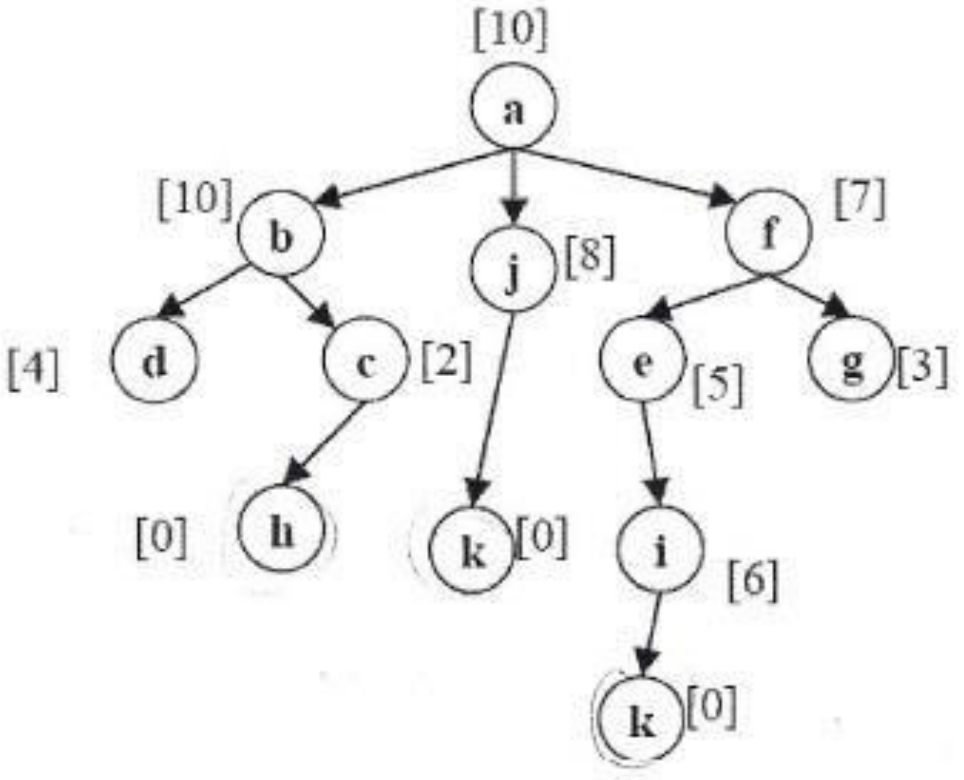
$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(f_7) - \text{value}(a_{10})$$

$$\text{value}(f_7) = -\text{heuristic}(f_7) = -7$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}



Check if next node f_7 is better than current node

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

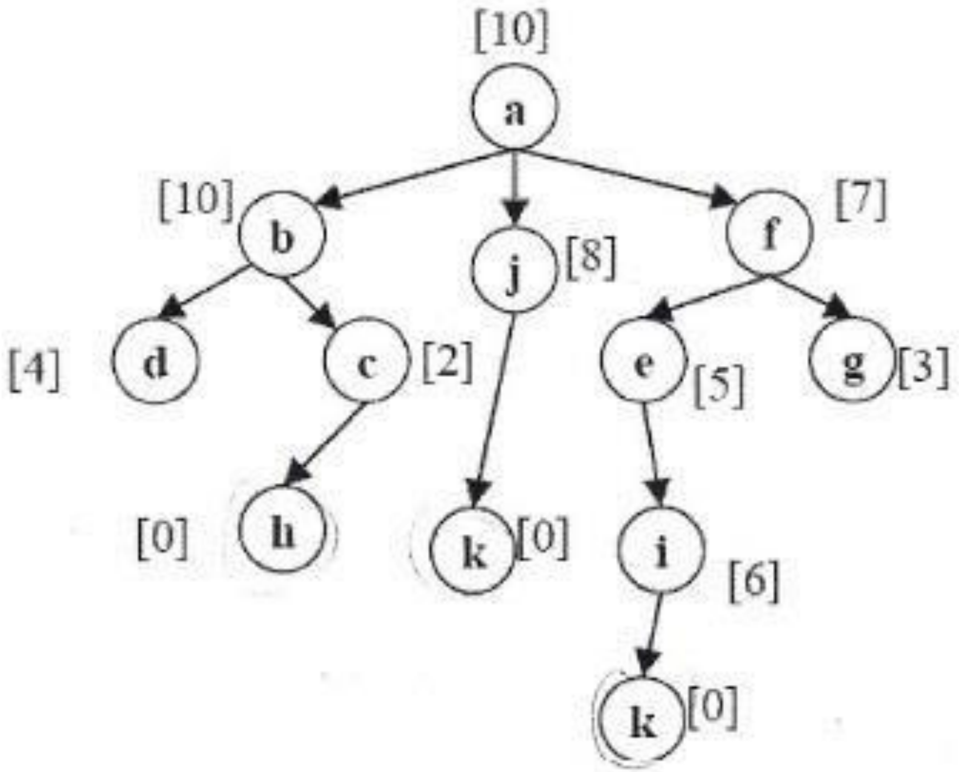
$$\Delta E = \text{value}(f_7) - \text{value}(a_{10})$$

$$\text{value}(f_7) = -\text{heuristic}(f_7) = -7$$

$$\text{value}(a_{10}) = -\text{heuristic}(a_{10}) = -10$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}



Check if next node f_7 is better than current node

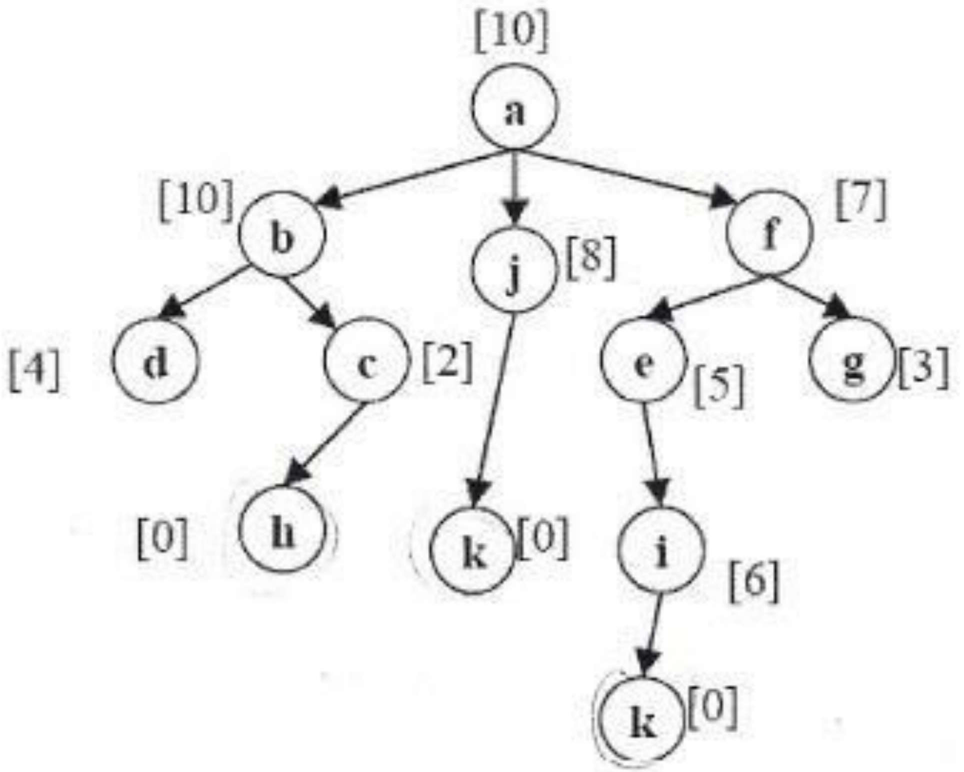
$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(f_7) - \text{value}(a_{10})$$

$$\Delta E = -7 - (-10) = +3$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}



Check if next node f_7 is better than current node

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(f_7) - \text{value}(a_{10})$$

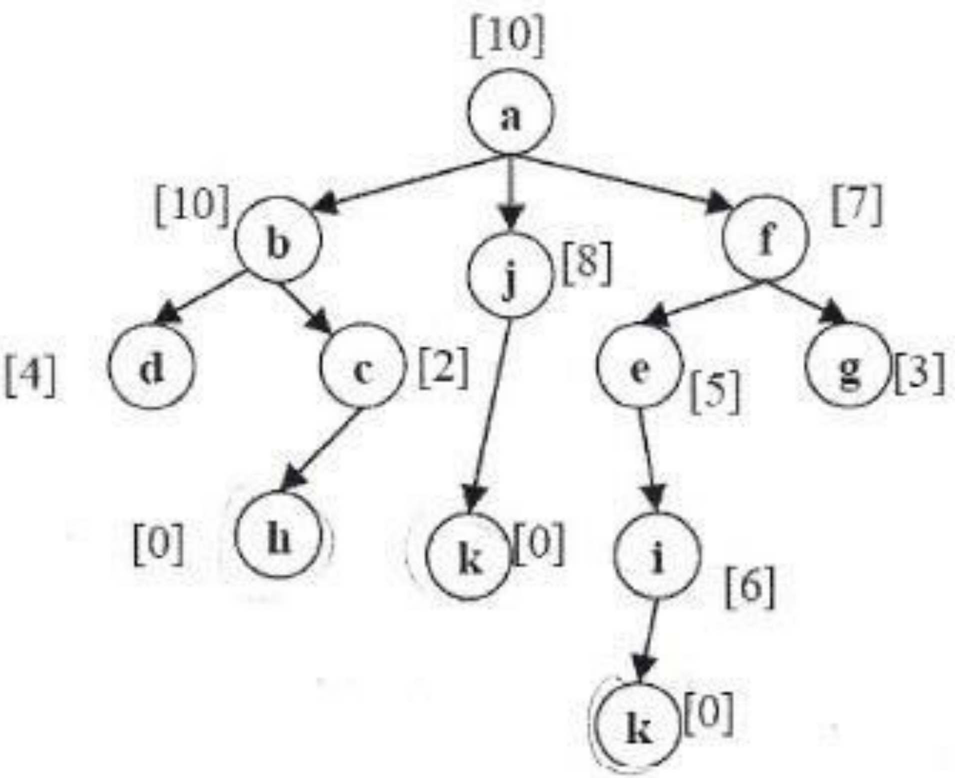
$$\Delta E = -7 - (-10) = +3$$

$$\because \Delta E > 0$$

$\therefore f_7$ **will be selected with probability 1**

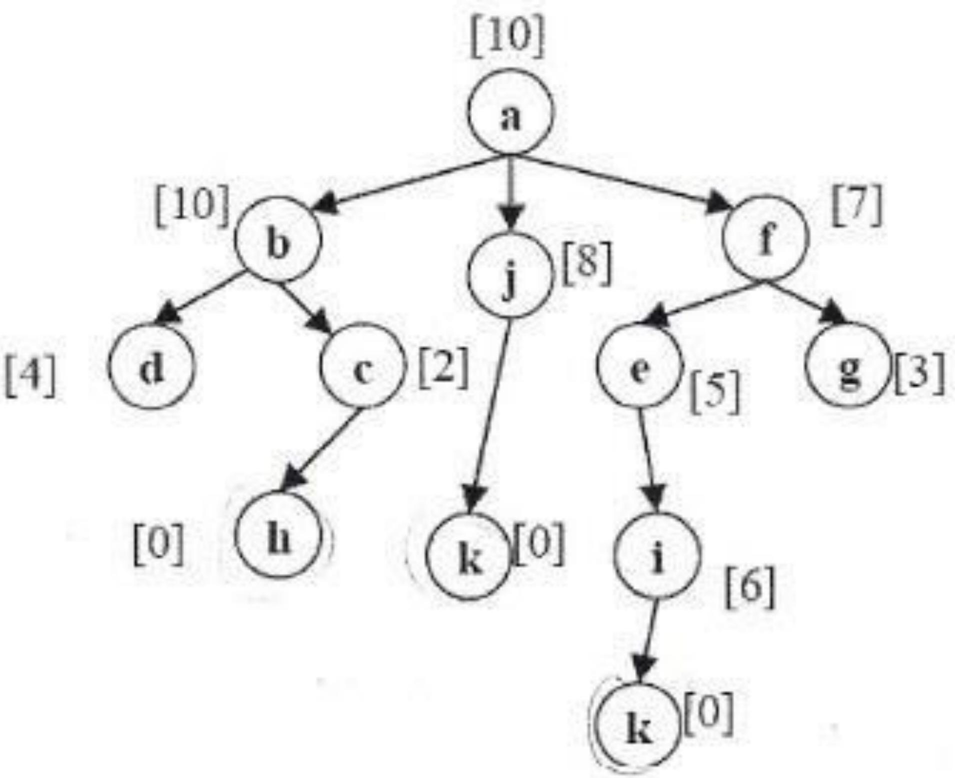
Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	



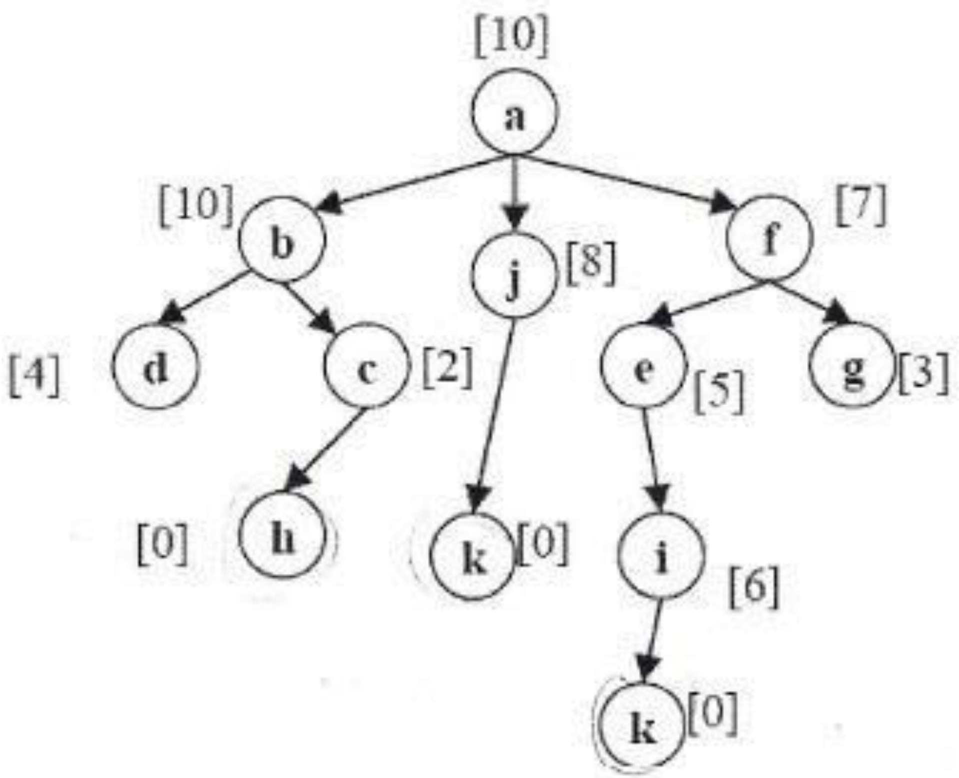
Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	



Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

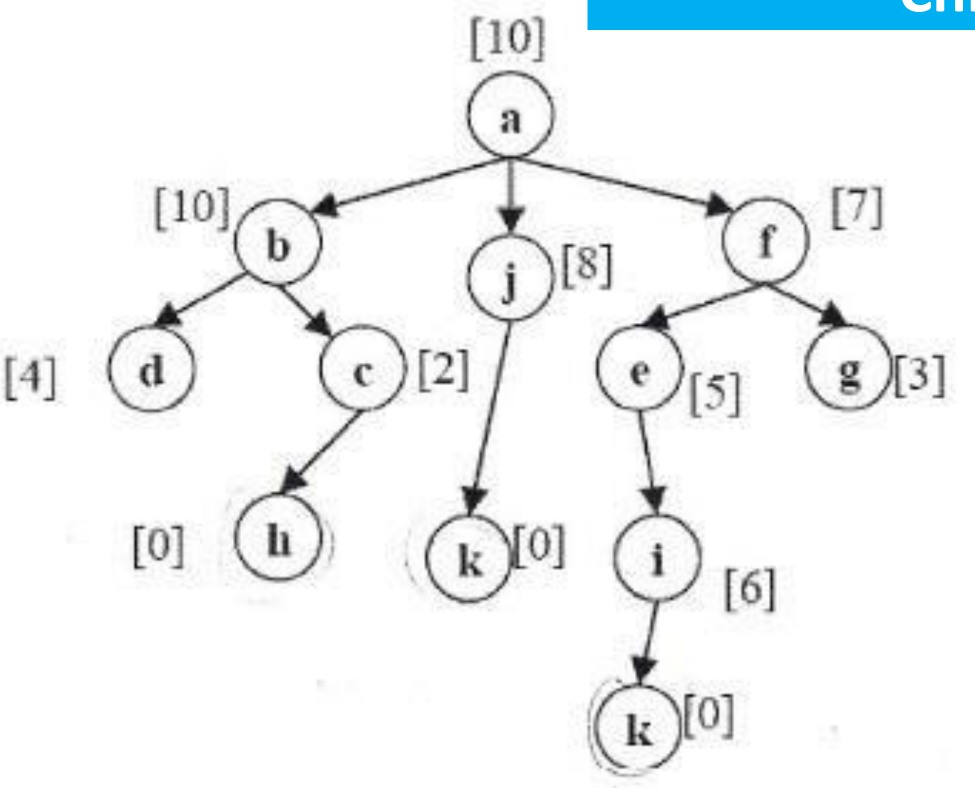


ccredited 'A' Grade
ESTD. 1963
9001:2015
st College
ITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

Randomly Select a Child

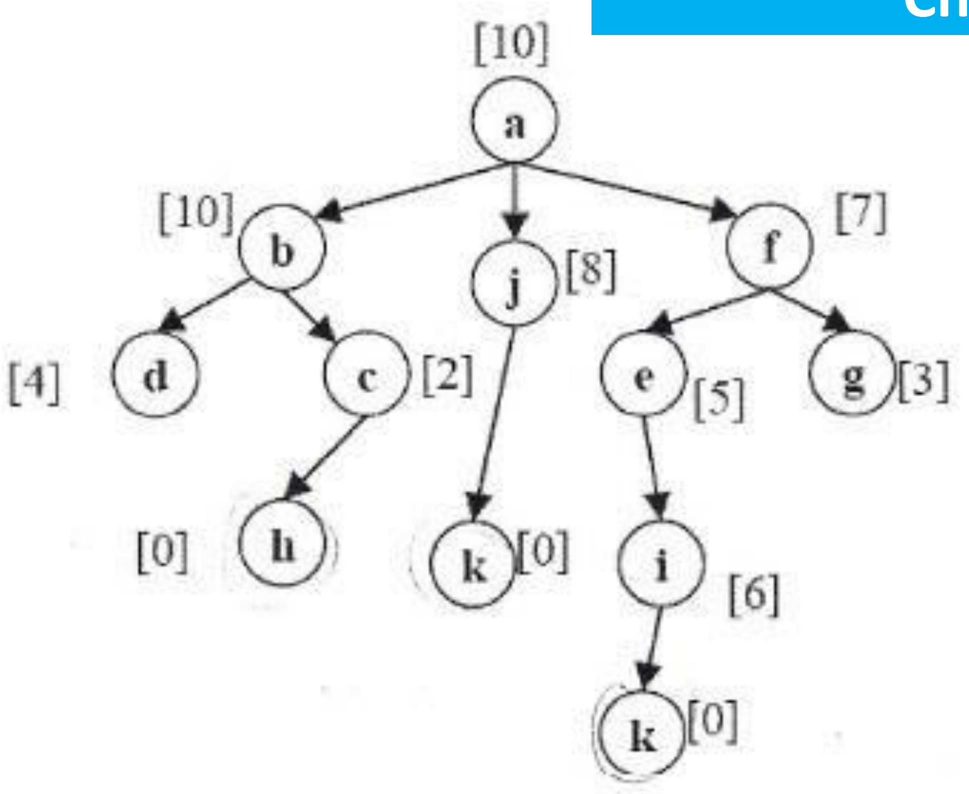


ESTD. 1963
9001:2015
st College
ITY OF MUMBAI

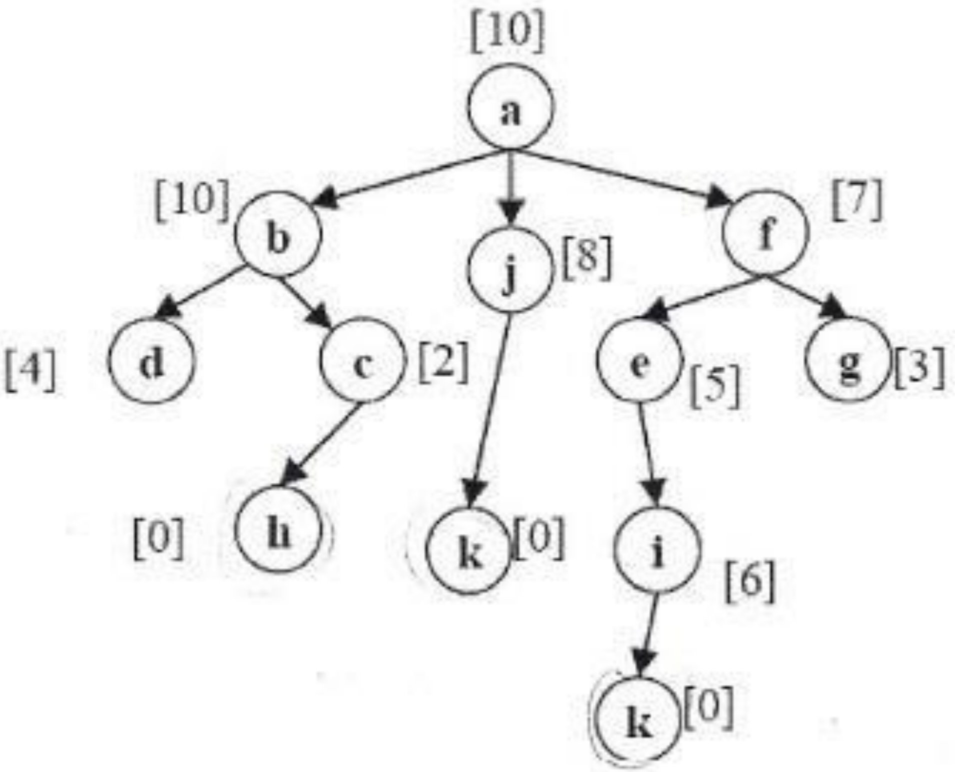
Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

Randomly Select a Child



Simulated Annealing

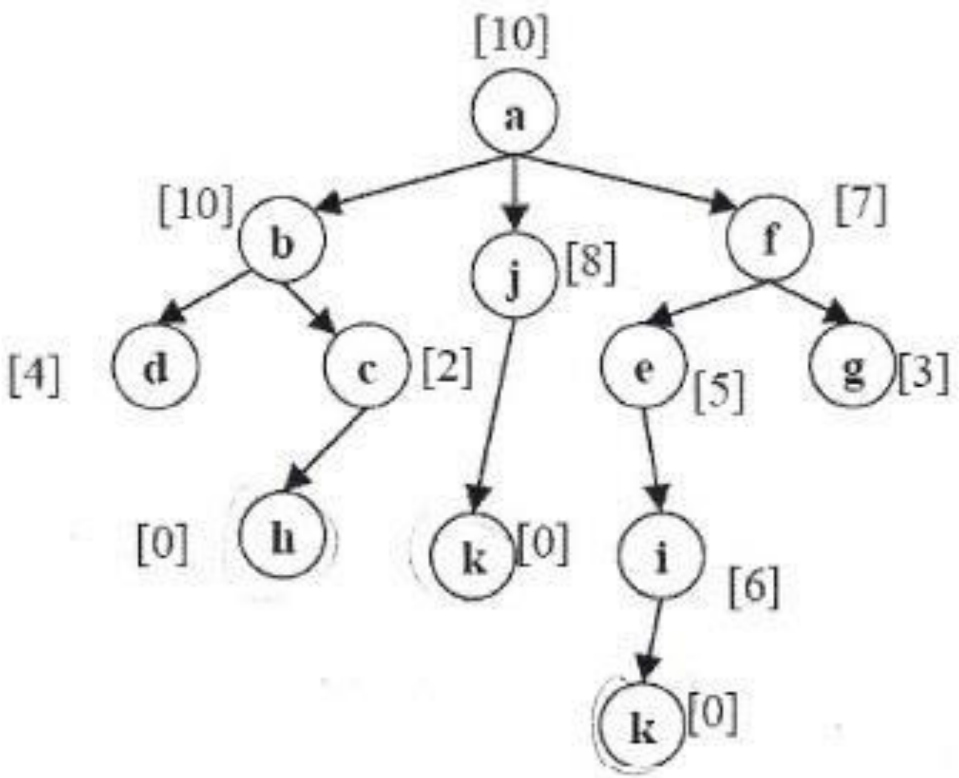


Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

Check if next node e_5 is better than current node

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

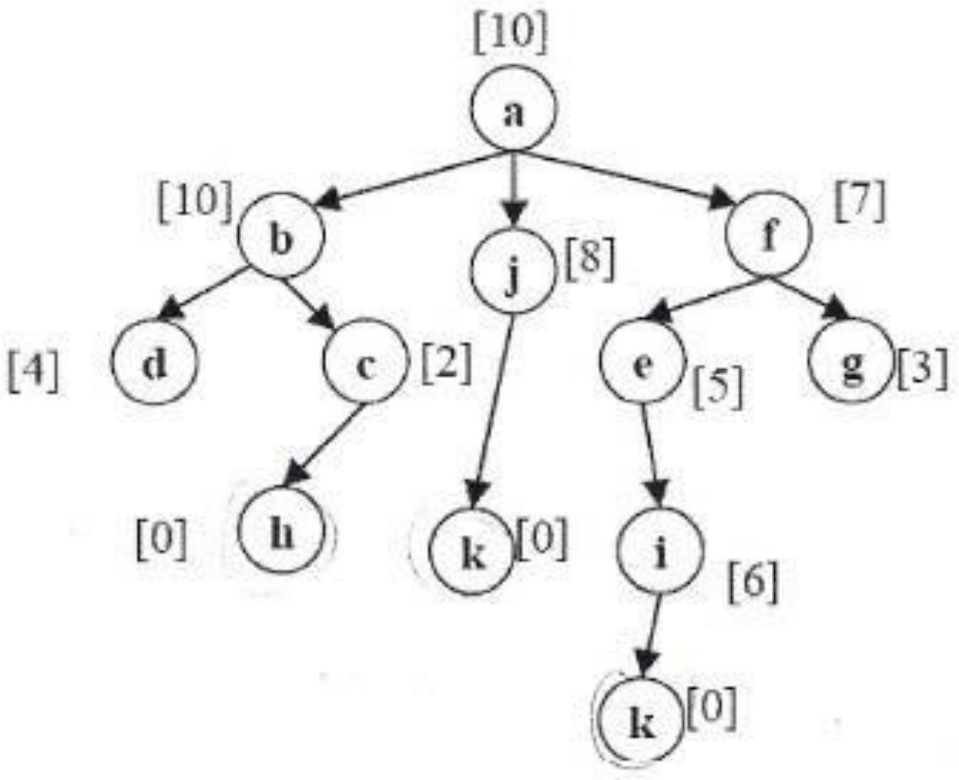


Check if next node e_5 is better than current node

$$\Delta E > 0$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3



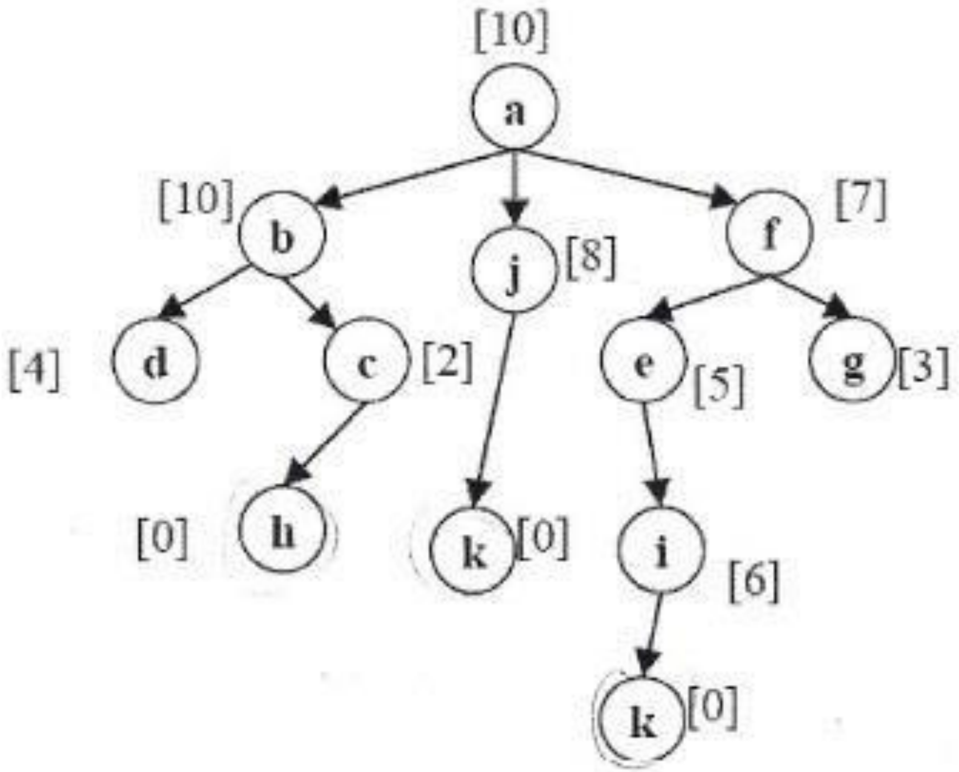
Check if next node $e5$ is better than current node

$$\Delta E = \text{value}(\text{ next }) - \text{value}(\text{current})$$

Accredited 'A' Grade ESTD. 1963 RJC 9001:2015 st College ITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3



Check if next node e_5 is better than current node

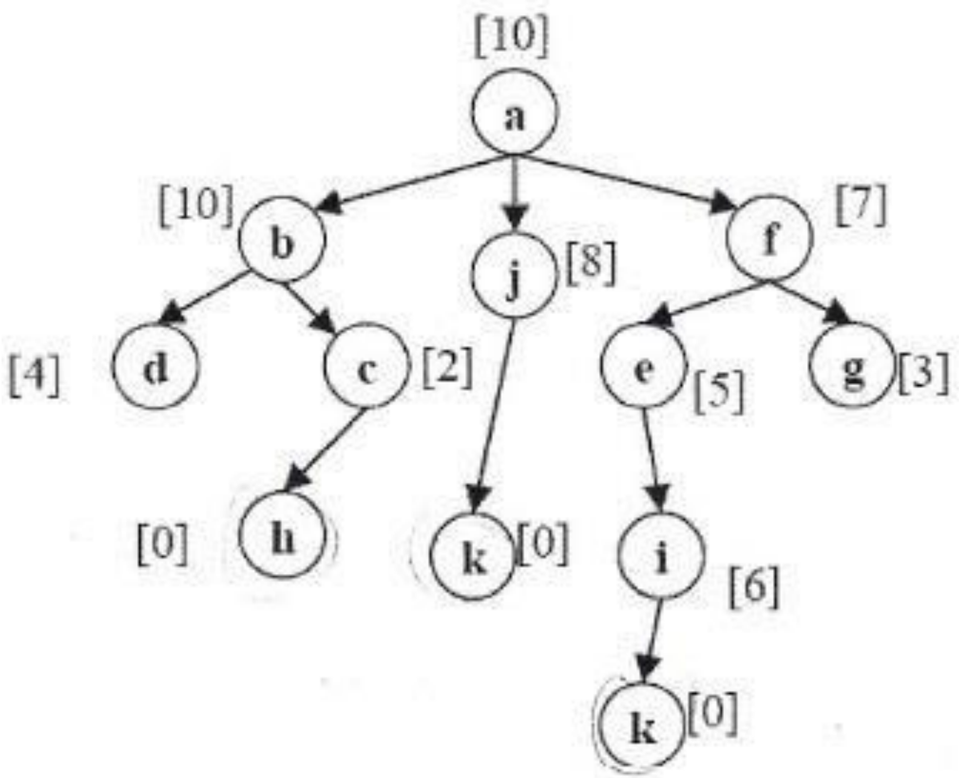
$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(e_5) - \text{value}(f_7)$$

Accredited 'A' Grade ESTD. 1963 RJC 9001:2015 ... st College ... ITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3



Check if next node e_5 is better than current node

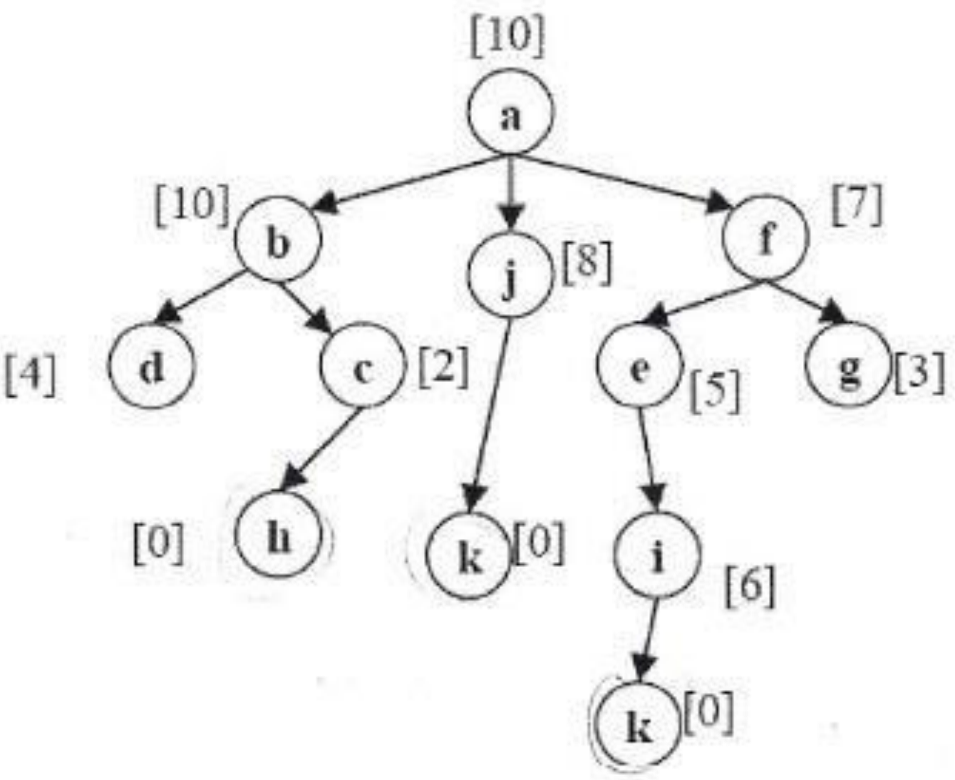
$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(e_5) - \text{value}(f_7)$$

$$\text{value}(e_5) = -\text{heuristic}(e_5) = -5$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3



Check if next node e_5 is better than current node

$$\Delta \mathrm{E} = \mathrm{value(next)} - \mathrm{value(current)}$$

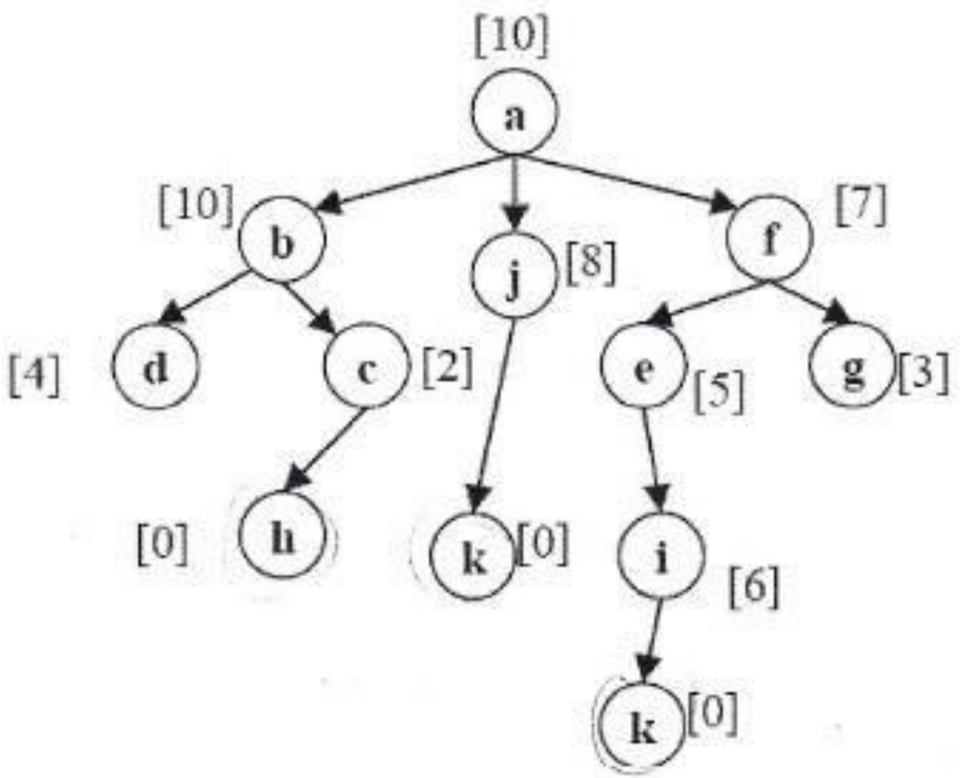
$$\Delta \mathrm{E} = \mathrm{value}(e_5) - \mathrm{value}(f_7)$$

$$\mathrm{value}(e_5) = -\mathrm{heuristic}(e_5) = -5$$

$$\mathrm{value}(f_7) = -\mathrm{heuristic}(f_7) = -7$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3



Check if next node e_5 is better than current node

$$\Delta E = \text{value(next)} - \text{value(current)}$$

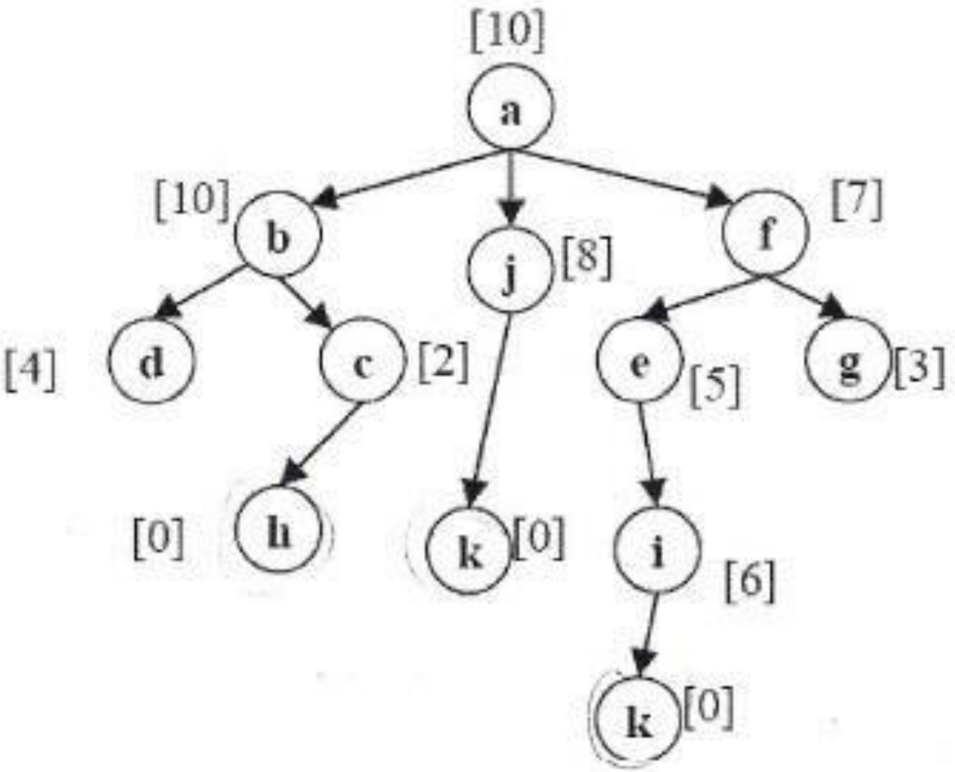
$$\Delta E = \text{value}(e_5) - \text{value}(f_7)$$

$$\Delta E = -5 - (-7) = +2$$

Simulated Annealing

$\because \Delta E > 0$

$\therefore e_5$ **will be selected with probability 1**



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3

Check if next node e_5 is better than current node

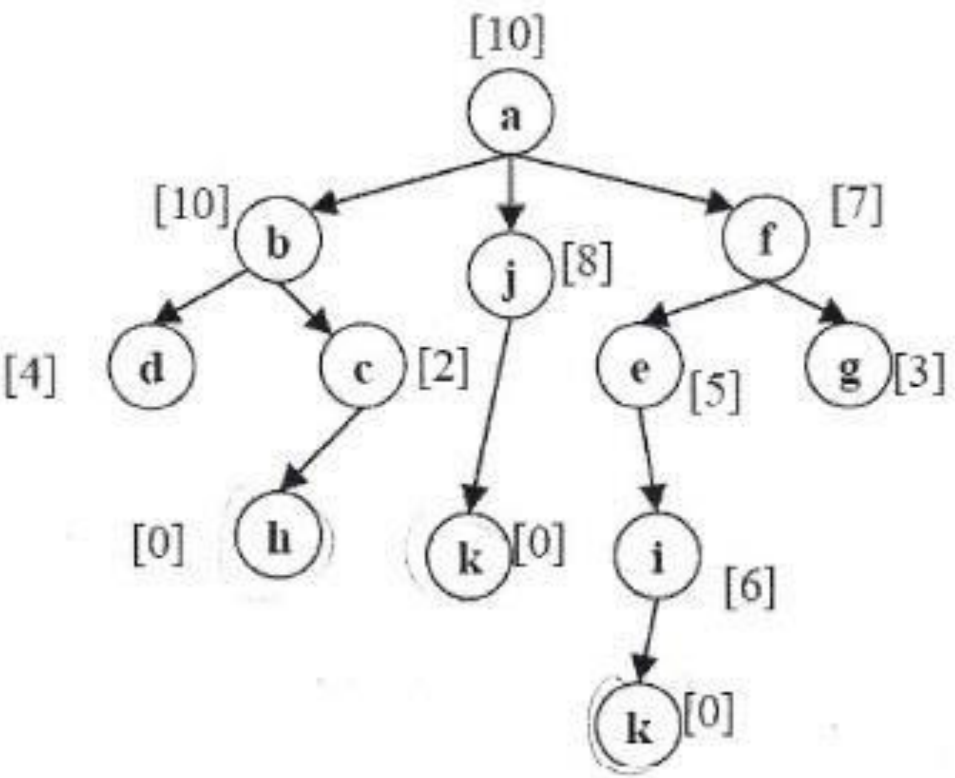
$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(e_5) - \text{value}(f_7)$$

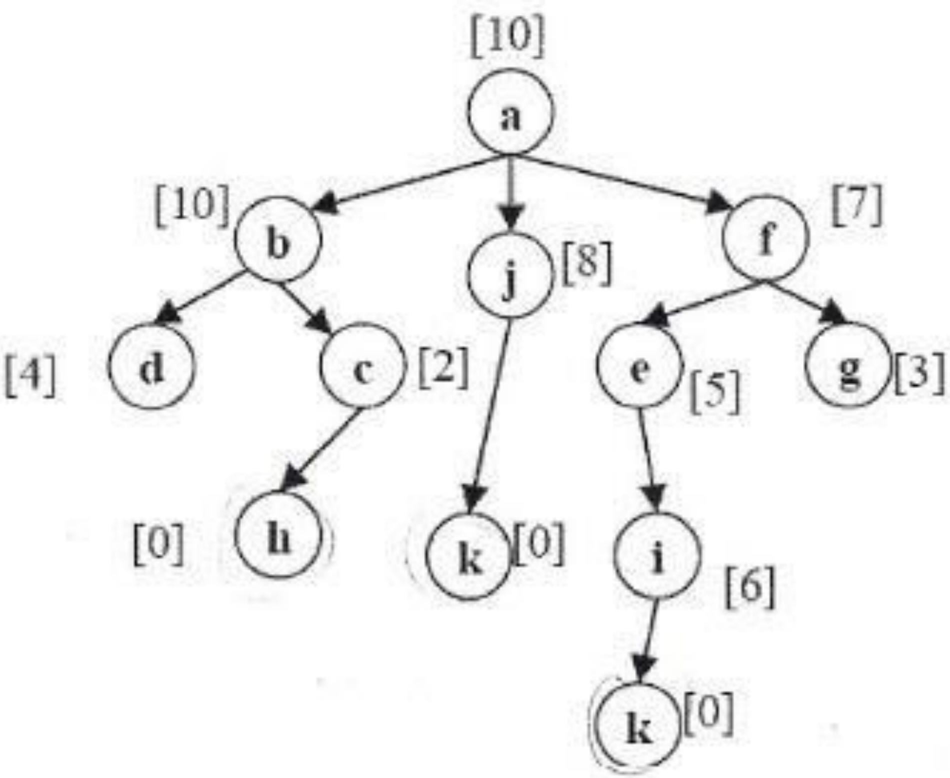
$$\Delta E = -5 - (-7) = +2$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	



Simulated Annealing

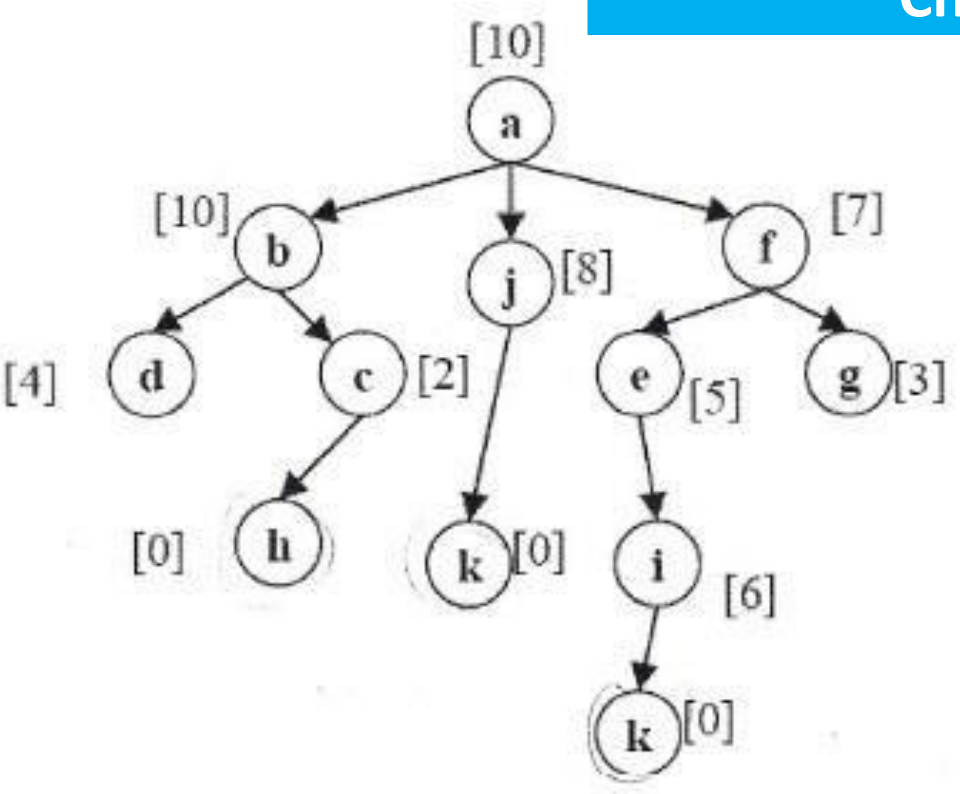


Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

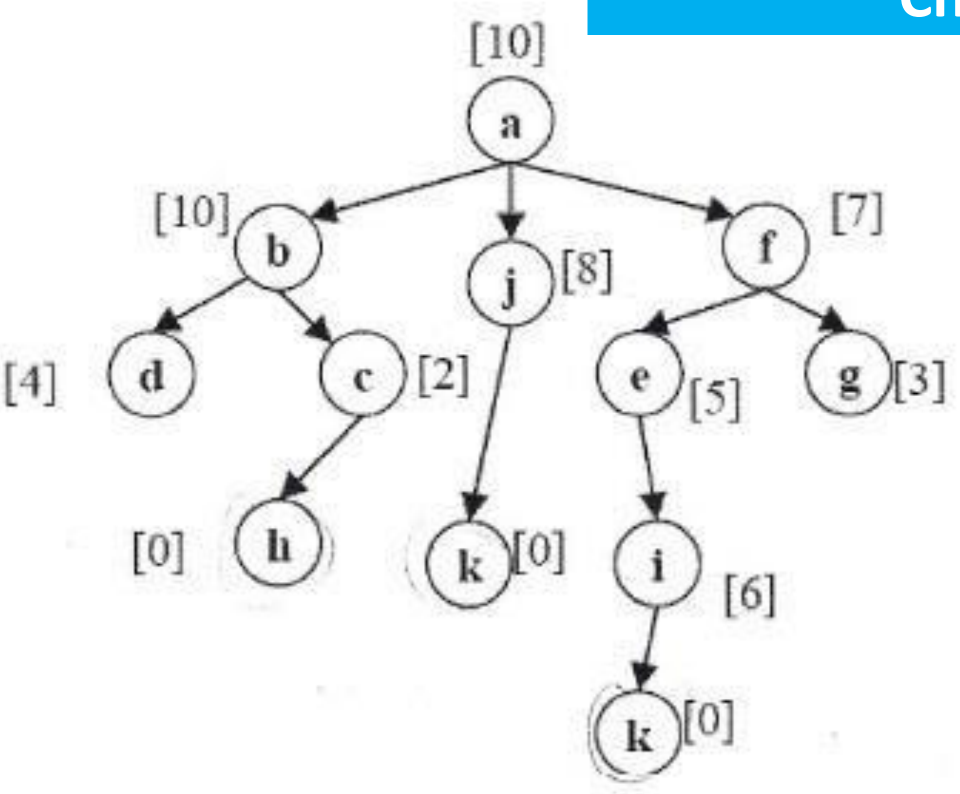
Randomly Select a Child



ESTD. 1963
Grade
9001:2015
st College
ITY OF MUMBAI

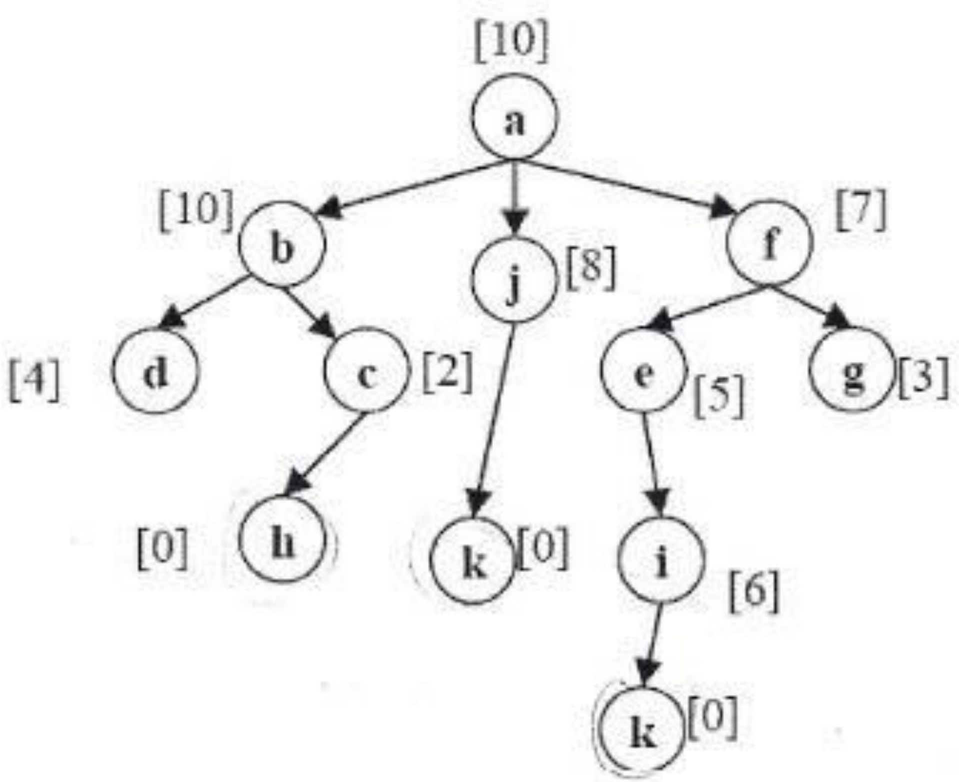
Simulated Annealing

Randomly Select a Child



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Simulated Annealing

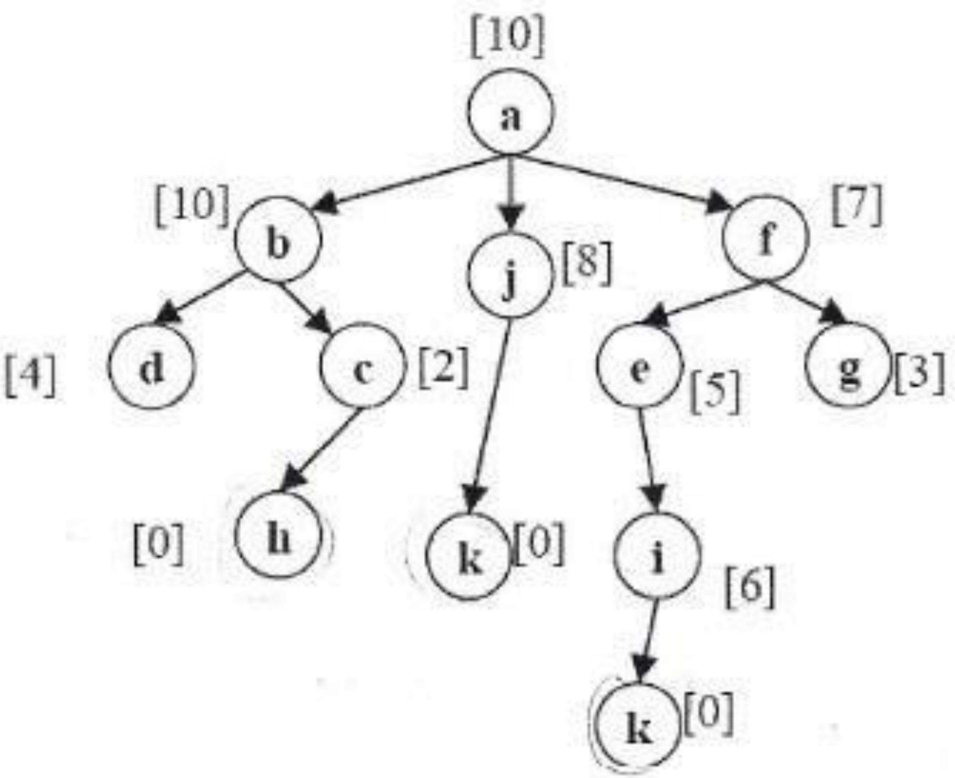


Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node i_6 is better than current node

ccredited 'A' Grade
ESTD. 1963
RJC
9001:2015
st College
ITY OF MUMBAI

Simulated Annealing

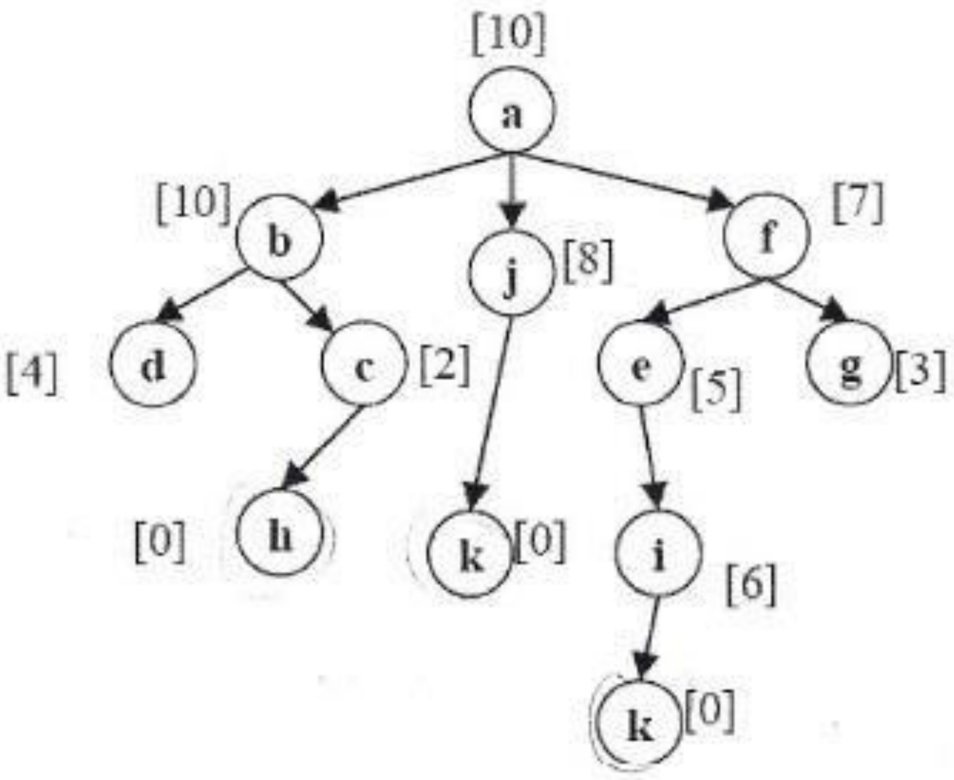


Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node i_6 is better than current node

$$\Delta E > 0$$

Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node i_6 is better than current node

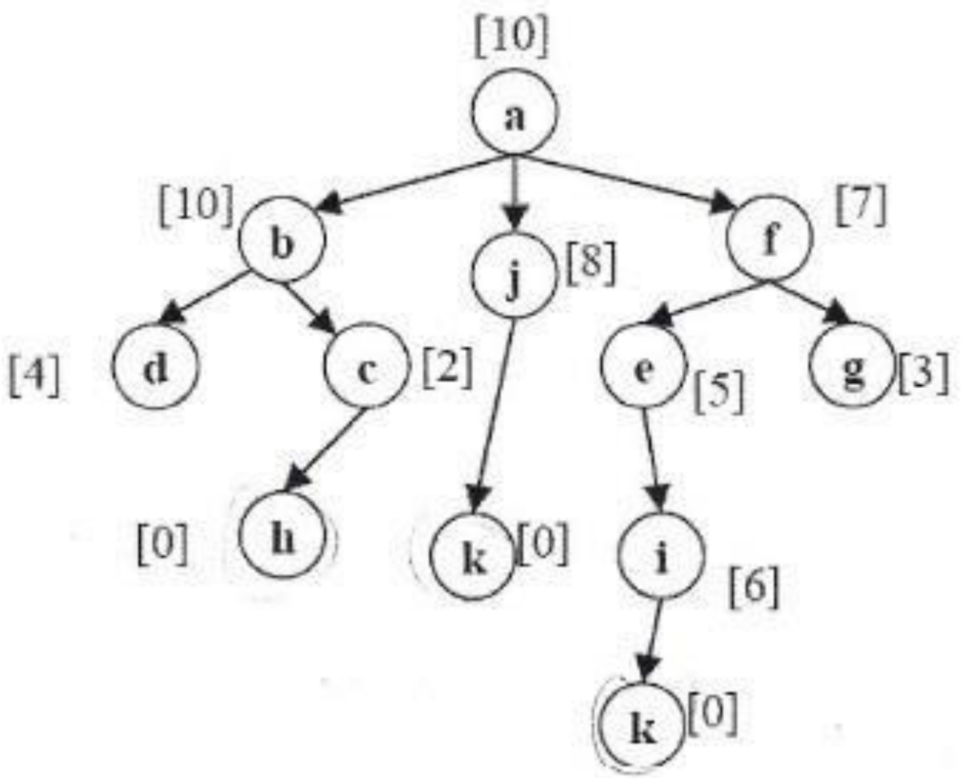
$$\Delta E > 0$$

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(i_6) - \text{value}(e_5)$$

Simulated Annealing

$$\text{value}(i_6) = -\text{heuristic}(i_6) = -6$$



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node i_6 is better than current node

$$\Delta E > 0$$

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

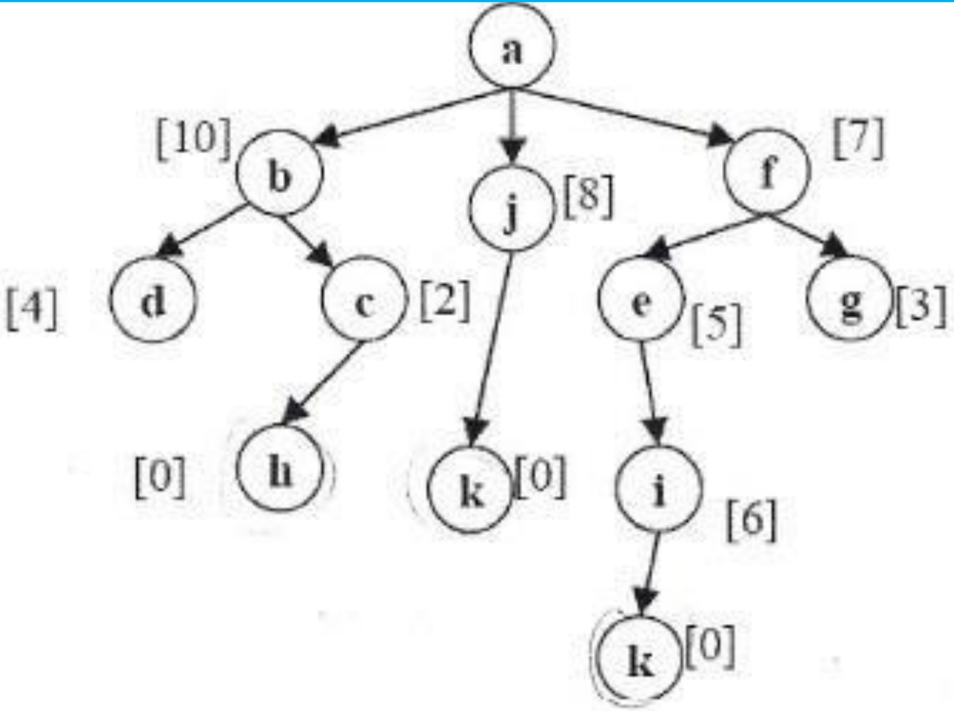
$$\Delta E = \text{value}(i_6) - \text{value}(e_5)$$

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

$$\text{value}(i_6) = -\text{heuristic}(i_6) = -6$$

$$\text{value}(e_5) = -\text{heuristic}(e_5) = -5$$



Check if next node i_6 is better than current node

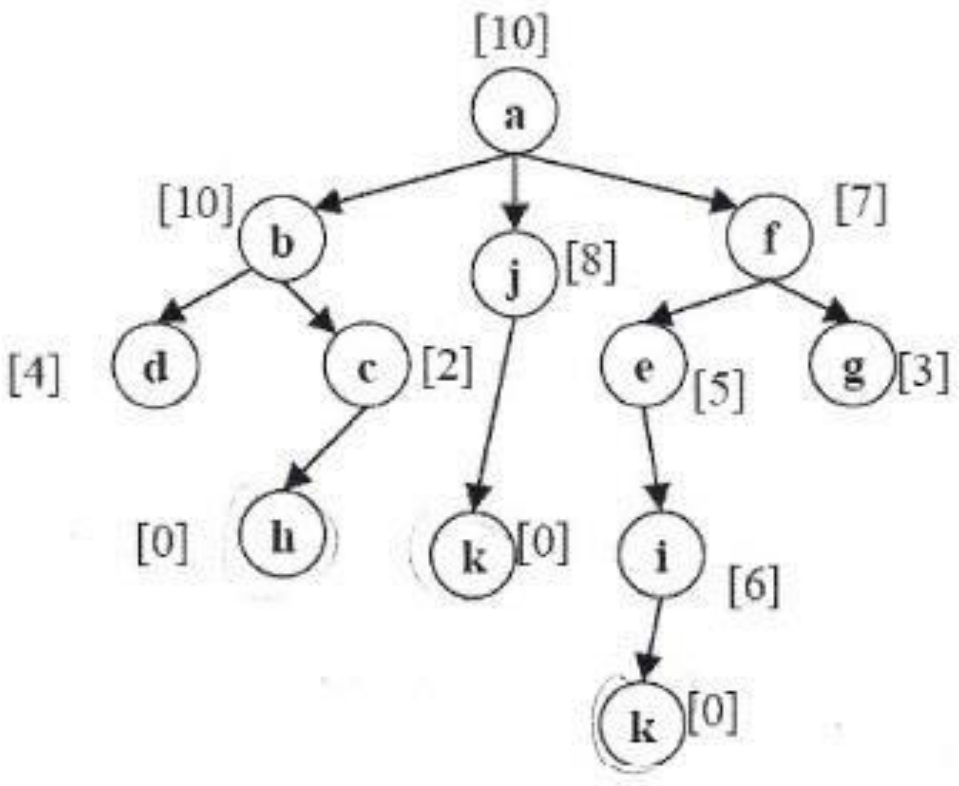
$$\Delta E > 0$$

$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(i_6) - \text{value}(e_5)$$

Simulated Annealing

$$\Delta E = -6 - (-5) = -1$$



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node i_6 is better than current node

$$\Delta E > 0$$

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(i_6) - \text{value}(e_5)$$

Simulated Annealing

$\because \Delta E < 0$

$\therefore i_6$ can be selected with

probability $p = e^{\frac{\Delta E}{T}}$

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

Check if next node e_5 is better than current node

$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(e_5) - \text{value}(f_7)$$

$$\Delta E = -5 - (-7) = +2$$

ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6

$\because \Delta E < 0$

$\therefore i_6$ can be selected with probability $p = e^{\frac{\Delta E}{T}}$

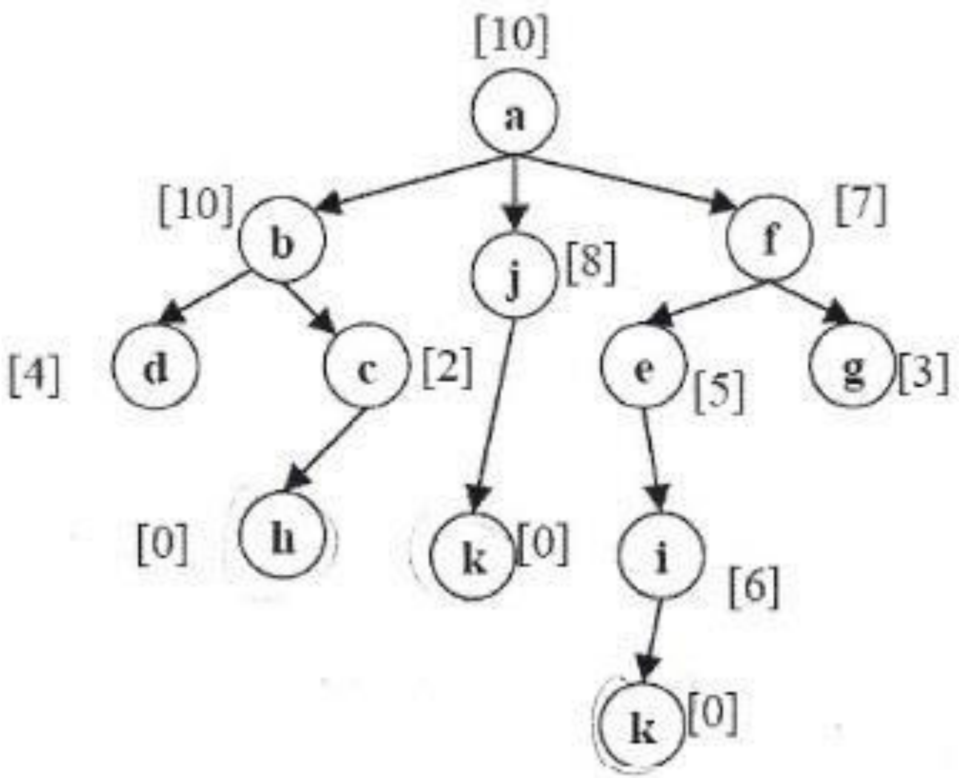
Check if next node e_5 is better than current node

$$p = e^{\frac{-1}{10}} = e^{\frac{-1}{10}} = .905$$

RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

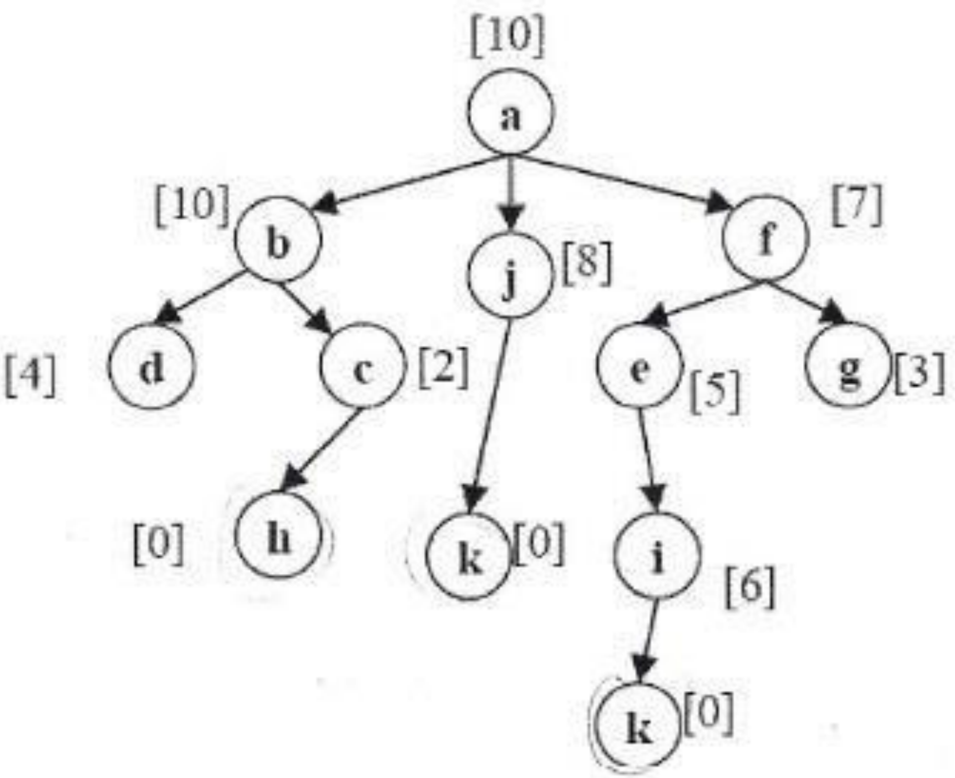
Simulated Annealing

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6



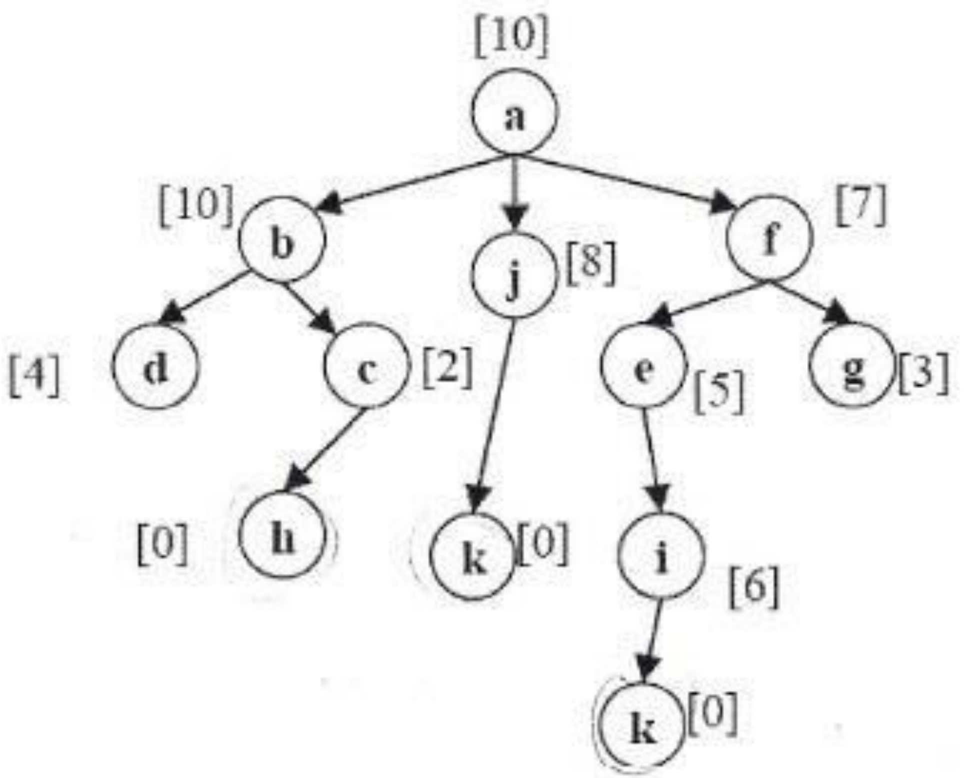
Because the only child of e_5 is i_6 then it will be selected even if its probability is not 1.

Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	

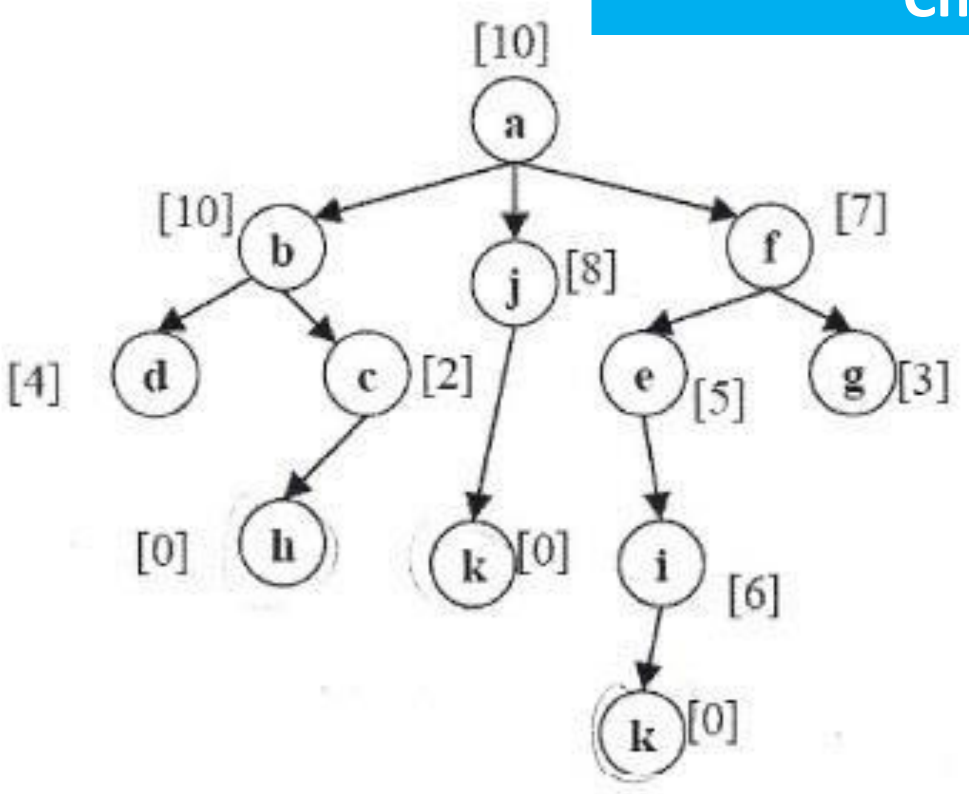
Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Simulated Annealing

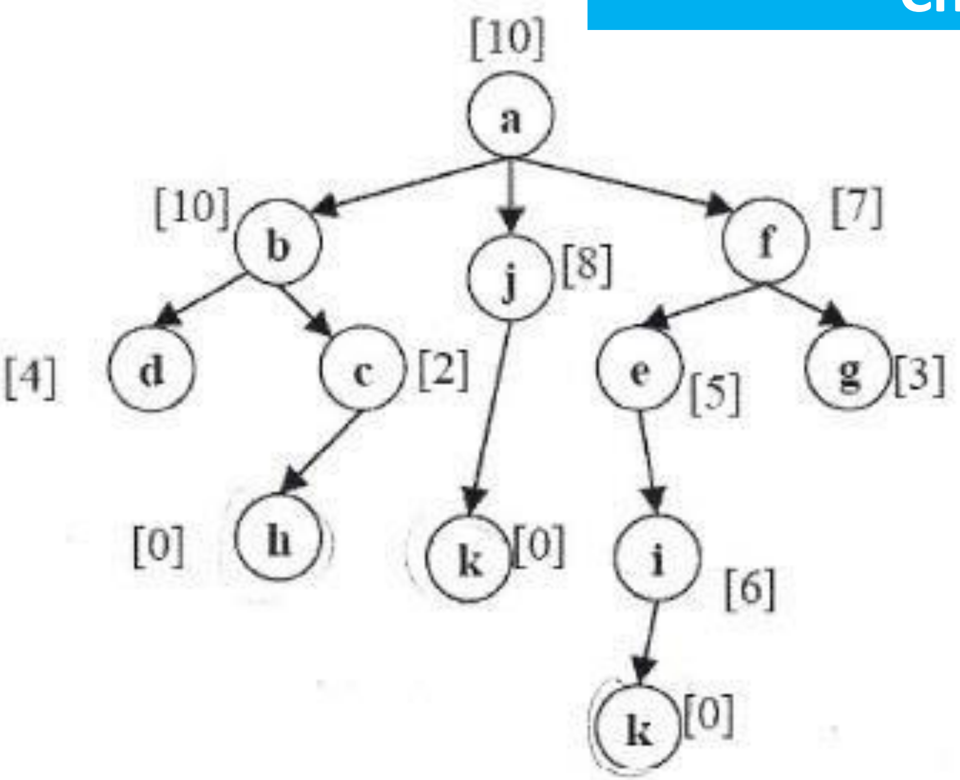
Randomly Select a Child



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

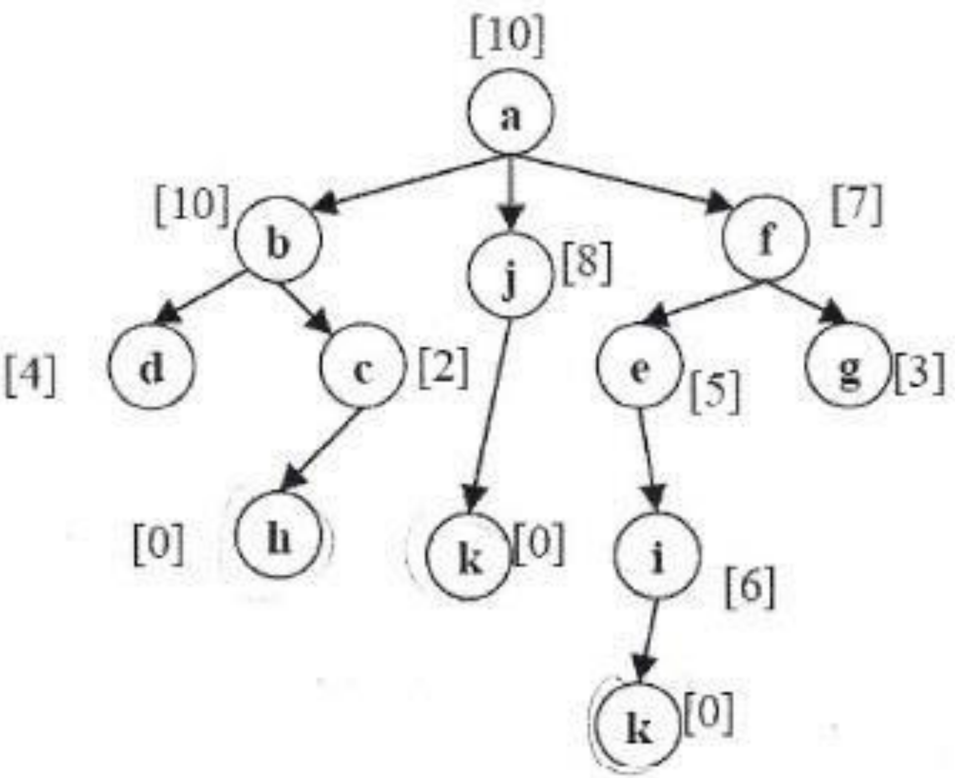
Simulated Annealing

Randomly Select a Child



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

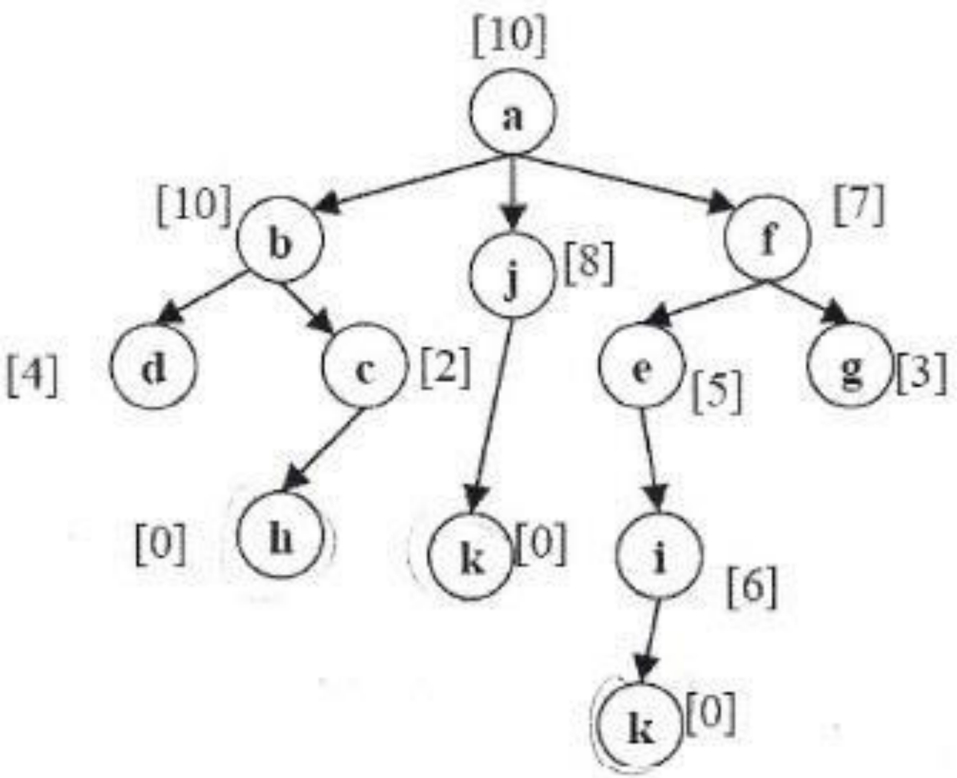
Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

Simulated Annealing



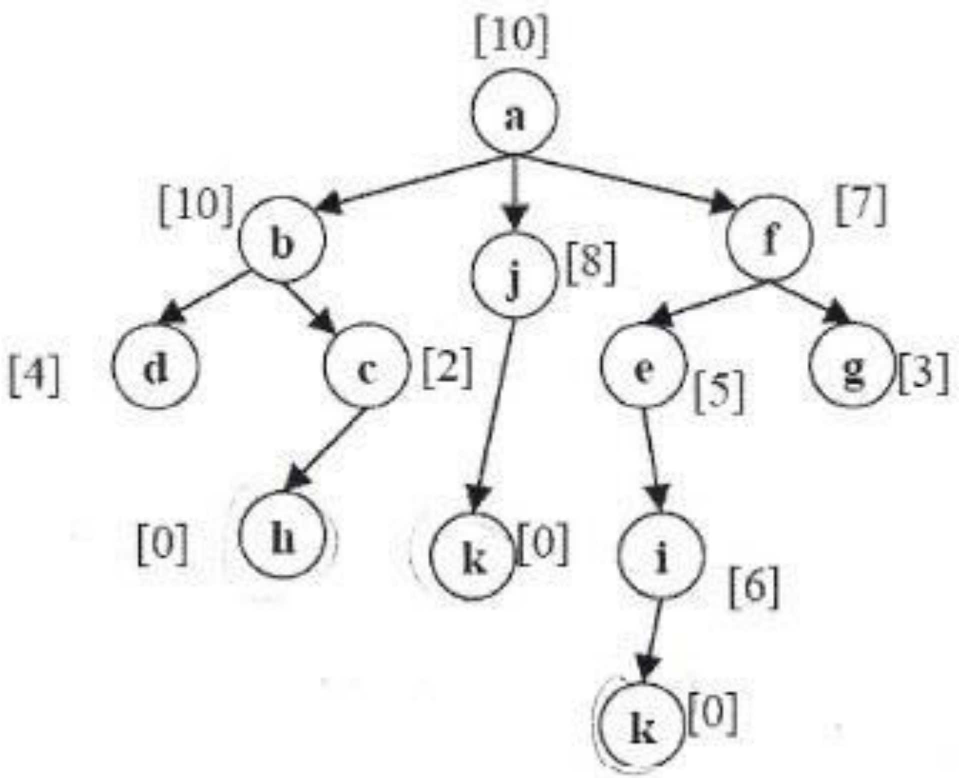
Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

$$\Delta E > 0$$

Simulated Annealing

$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

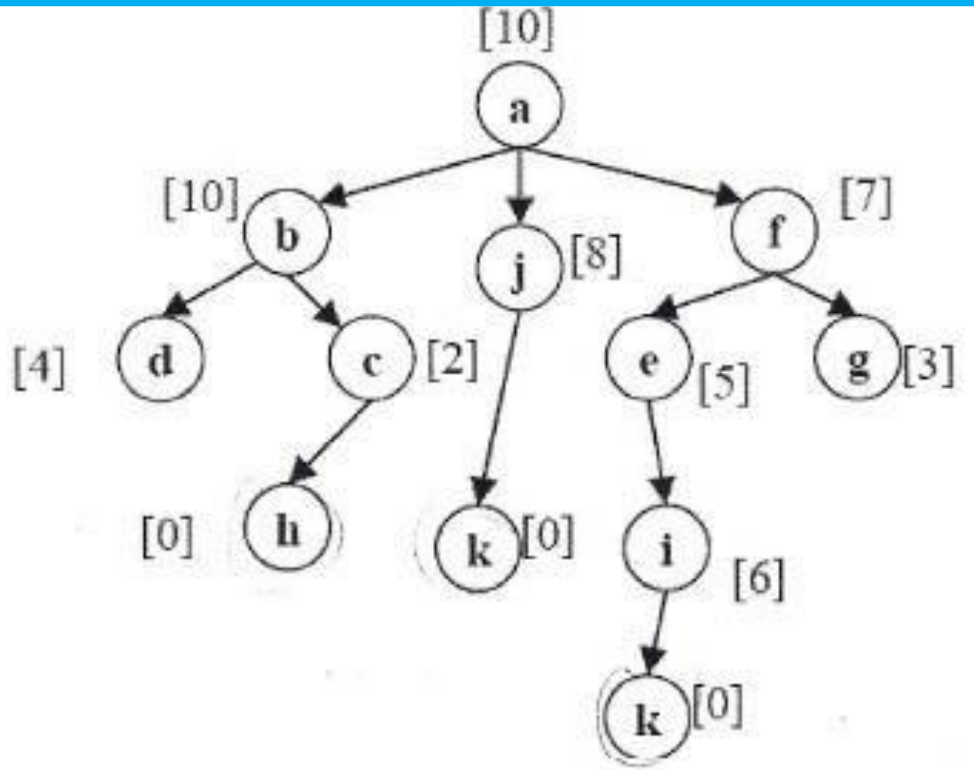
Check if next node k_0 is better than current node

$\Delta E > 0$

Simulated Annealing

$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(k_0) - \text{value}(i_6)$$



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

$$\Delta E > 0$$

ESTD. 1963 'A' Grade RJC 9001:2015 st College ITY OF MUMBAI

Simulated Annealing

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(k_0) - \text{value}(i_6)$$

$$\text{value}(k_0) = -\text{heuristic}(k_0) = 0$$

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

$$\Delta E > 0$$

ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Simulated Annealing

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(k_0) - \text{value}(i_6)$$

$$\text{value}(k_0) = -\text{heuristic}(k_0) = 0$$

$$\text{value}(i_6) = -\text{heuristic}(i_6) = -6$$

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

$$\Delta E > 0$$

ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Simulated Annealing

$$\Delta E = \text{value(next)} - \text{value(current)}$$

$$\Delta E = \text{value}(k_0) - \text{value}(i_6)$$

$$\text{value}(k_0) = -\text{heuristic}(k_0) = 0$$

$$\text{value}(i_6) = -\text{heuristic}(i_6) = -6$$

$$\Delta E = 0 - (-6) = +6$$

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

$$\Delta E > 0$$

Simulated Annealing

$$\Delta E = \text{value}(\text{next}) - \text{value}(\text{current})$$

$$\Delta E = \text{value}(k_0) - \text{value}(i_6)$$

$$\text{value}(k_0) = -\text{heuristic}(k_0) = 0$$

$$\text{value}(i_6) = -\text{heuristic}(i_6) = -6$$

$$\Delta E = 0 - (-6) = +6$$

$\because \Delta E > 0$

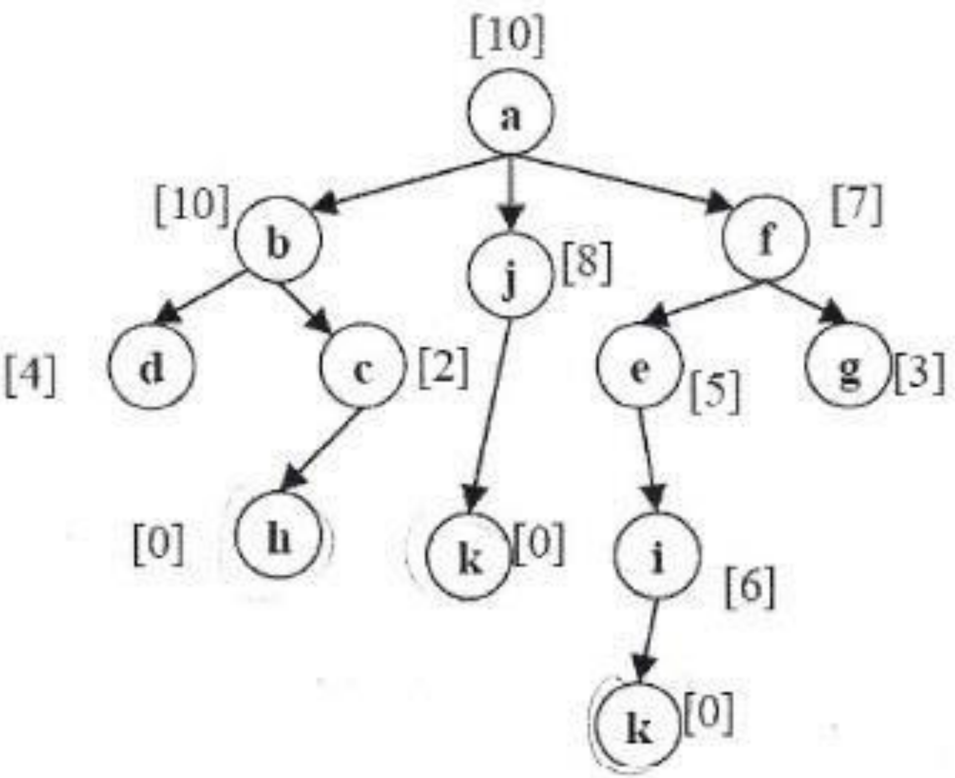
$\therefore k_0$ **will be selected with probability 1**

Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0

Check if next node k_0 is better than current node

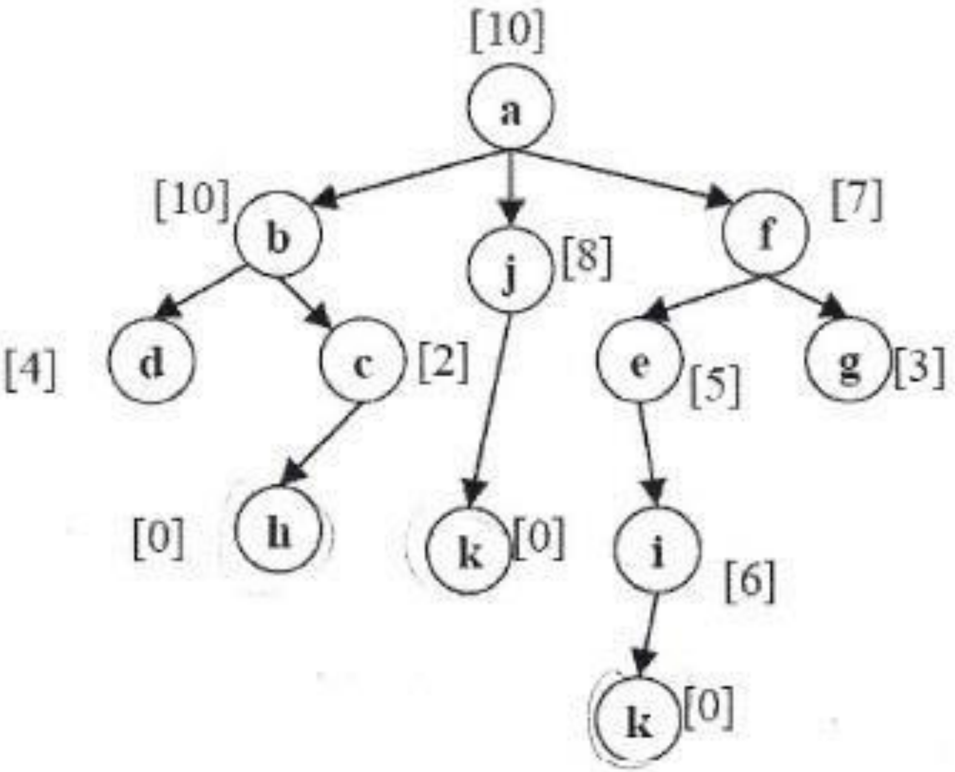
$$\Delta E > 0$$

Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0
k	

Simulated Annealing



Current	Children
a	---
a	f_7, j_8, b_{10}
f	e_5, g_3
e	i_6
i	k_0
k	

GOAL

Advantages & Disadvantages of Simulated Annealing

Advantages

- Can deal with arbitrary systems and cost functions
- Statistically guarantees finding an optimal solution
- It is relatively easy to code, even for complex problems
- Generally gives a ``good" solution

Disadvantages

- Repeatedly annealing schedule is very slow
- For problems where the energy landscape is smooth, or there are few local minima, SA is overkill
- The method cannot tell whether it has found an optimal solution. Some other method (e.g. branch and bound) is required to do this

NAAC Re-Accredited 'A' Grade
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

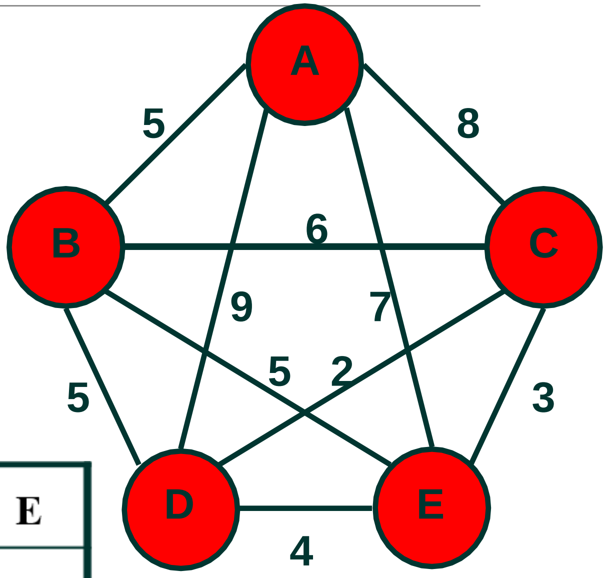
Uses of Simulated Annealing

- Used to solve combinatorial problems
- Applied to the travelling salesman problem to minimize the length of a route
- Formulation of power system optimization

NAAC Re-Accredited 'A' Grade
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Travelling Salesman Problem

- A salesman wants to visit a list of cities
  - ➲ Stopping in each city only once
  - ➲ Returning to the first city
  - ➲ Traveling the shortest distance
- Nodes are cities
- Arcs are labeled with distances between cities



	A	B	C	D	E
A	0	5	8	9	7
B	5	0	6	5	5
C	8	6	0	2	3
D	9	5	2	0	4
E	7	5	3	4	0

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MU

- **A solution is a permutation of cities, called a tour**
- How many solutions exist?
  - ➲ (n-1)!/2 where n = # of cities

n = 5 results in 12 tours

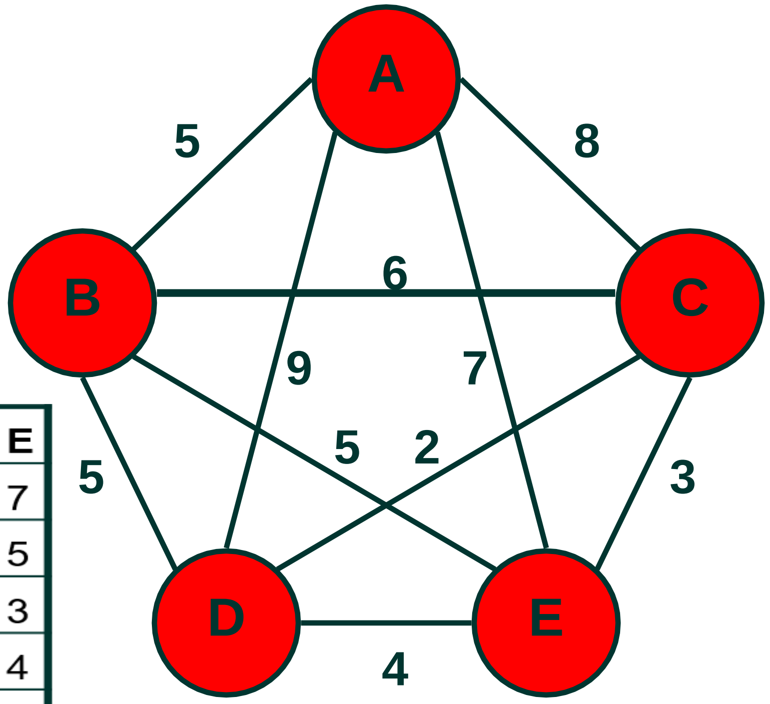
n = 10 results in 181440 tours

n = 20 results in ~$6*10^{16}$ tours

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY

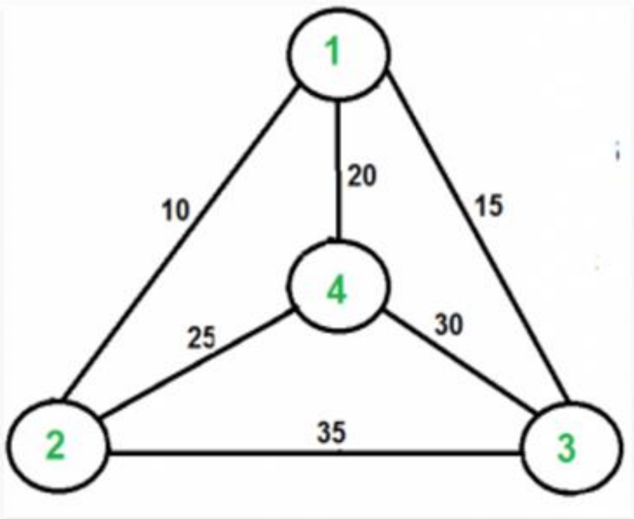
	A	B	C	D	E
A	0	5	8	9	7
B	5	0	6	5	5
C	8	6	0	2	3
D	9	5	2	0	4
E	7	5	3	4	0

5 City TSP



Naive Approach

- Consider city 1 as the starting and ending point. Since the route is cyclic, we can consider any point as a starting point.
- Generate all (n-1)! permutations of cities.
- Calculate the cost of every permutation and keep track of the minimum cost permutation.
- Return the permutation with minimum cost.



	1	2	3	4
1	0	10	15	20
2	10	0	35	25
3	15	35	0	30
4	20	25	30	0

NAAC Re-Accre... Grade
विद्या अ...
ESTD
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

graph = [[0, 10, 15, 20], [10, 0, 35, 25],[15, 35, 0, 30], [20, 25, 30, 0]]

Permutations:

1-2-3-4-1 -> Cost = 95 , 1-4-2-3-1 ->Cost = 95 … . .

1-3-4-2-1 -> Cost = 80

tsp2.py - D:\python\tsp2.py (3.9.2)

File Edit Format Run Options Window Help

```python
# Python program to implement traveling salesman
# problem using naive approach.
from sys import maxsize
from itertools import permutations
V = 4

# implementation of traveling Salesman Problem
def travellingSalesmanProblem(graph, s):

        # store all vertex apart from source vertex
        vertex = []
        for i in range(V):
                if i != s:
                        vertex.append(i)

        # store minimum weight Hamiltonian Cycle
        min_path = maxsize
        next_permutation=permutations(vertex)
        for i in next_permutation:

                # store current Path weight(cost)
                current_pathweight = 0

                # compute current path weight
                k = s
                for j in i:
                        current_pathweight += graph[k][j]
                        k = j
                current_pathweight += graph[k][s]

                # update minimum
                min_path = min(min_path, current_pathweight)

        return min_path
```

tsp2.py - D:\python\tsp2.py (3.9.2)

File Edit Format Run Options Window Help

```python
def next_permutation(L):
    ''' Permute the list L in-place to generate the next lexicographic permutation.
    Return True if such a permutation exists, else return False.
    '''
    n = len(L)

    # Step 1: find rightmost position i such that L[i] < L[i+1]
    i = n - 2
    while i >= 0 and L[i] >= L[i+1]:
        i -= 1

    if i == -1:
        return False

    # Step 2: find rightmost position j to the right of i such that L[j] > L[i]
    j = i + 1
    while j < n and L[j] > L[i]:
        j += 1
    j -= 1

    # Step 3: swap L[i] and L[j]
    L[i], L[j] = L[j], L[i]


    # Step 4: reverse everything to the right of i
    left = i + 1
    right = n - 1

    while left < right:
        L[left], L[right] = L[right], L[left]
        left += 1
        right -= 1
    return True
```

```python
# Main Code
if __name__ == "__main__":

	# matrix representation of graph
	graph = [[0, 10, 15, 20], [10, 0, 35, 25],[15, 35, 0, 30], [20, 25, 30, 0]]
	s = 0
	print(travellingSalesmanProblem(graph, s))
```

IDLE Shell 3.9.2

File Edit Shell Debug Options Window Help

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========================== RESTART: D:\python\tsp2.py ==========================
80
>>> |
```

References

Web references:

- https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_popular_search_algorithms.htm
- https://www.goeduhub.com/7743/water-jug-problem-in-artificial-intelligence
- http://classes.engr.oregonstate.edu/eecs/spring2018/cs331/slides/LocalSearch.2pp.pdf
- Book:
- **Artificial Intelligence**: A Modern Approach · Third edition, Stuart **Russell** and Peter Norvig

NAAC Re-Accredited 'A' Grade
विद्यया अमृतं अश्नुते
ESTD. 1963
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI

Assignment

- **Problem**: Given 3 jugs of capacities 8, 5 and 3 liters respectively. These are initially filled with 8, 0 and 0 liters. In the goal state they should be filled with 4, 4 and 0 liters.(There is no marking in the jugs)
- Write solution steps and python code to solve the above problem
- Prepare a Case Study on :Use of Beam search algorithm in Translation
- Solving Travelling Salesman Problem using dynamic Approach .Explain with a simple example

NAAC Re-Accredited 'A' Grade
विद्या अमृतं अश्नुते
ESTD. 1963
RJC
ISO 9001:2015
Best College
UNIVERSITY OF MUMBAI