Heuristic algorithms

→ **Heuristic:**

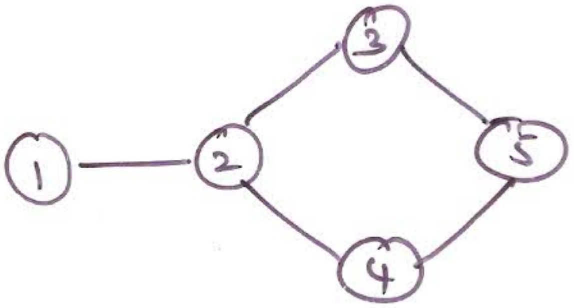
- A procedure that may produce a good or even optimal solution to your problem if you are lucky, but that on the other hand may produce no solution or one that is far from optimal if you are not.
- May be deterministic or probabilistic

→ **Approximation algorithm:**

- Always provides some kind of solution to your problem though it may fail to find the optimal solution.

I Colouring a graph:

- $G = \langle N, A \rangle$ be an undirected graph.
- To paint the nodes of G in such a way that no two adjacent nodes are the same colour.
- To find : the minimum no. of colours required.
- This minimum is called the chromatic number of the graph.



→ NP-Hard problem.

R - 1, 3, 4,   B - 2, 5

Greedy Heuristic:

- choose a colour and an arbitrary starting node and consider each node in turn.
- If a node can be painted with the first color (we do so)

- When no further nodes can be painted, we choose a new colour and a new starting node that has not yet been painted.
- Then paint as many nodes as we can with the second colour.
- so on...

Scenario-1:

- 1 - R , 3, 4 - R
  2 - B , 5 - B } k = 2 (optimal solution)

Scenario-2:

Consider nodes in the order 1, 5, 2, 3, 4

1, 5 - R   2 - B , 3, 4 - requires a third colour

→ k = 3 (not optimal)

→ This heuristic may not find an optimal solution, but we hope that it will find a "good" solution (not too different from the optimum).

→ For any graph 'G', there is atleast one ordering of the nodes, that allows the greedy algorithm to find an optimal solution.

→ Consider any graph 'G' and suppose that an optimal solution requires 'k' colours. You are given a way of colouring 'G' just using k-colours.

(i) Number these 'k' colours arbitrarily

(ii) Number the nodes of 'G' as follows:

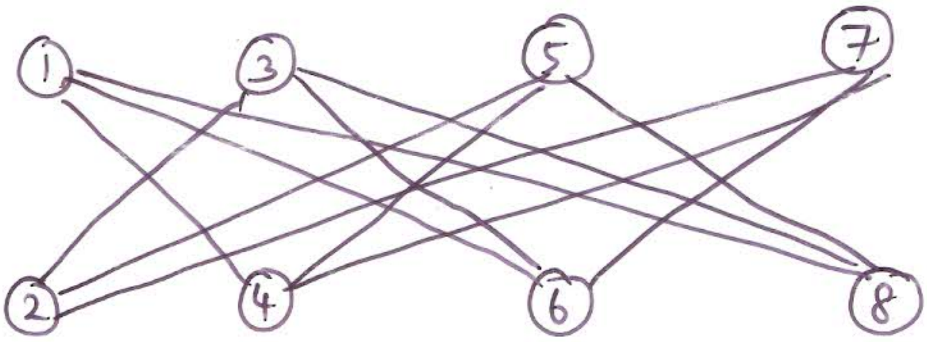
- first number consecutively all the nodes of 'G' that are painted with colour 1
- Continue the sequence by numbering all those nodes that are painted with colour 2

- When you finish colour k, all the nodes will have been numbered.

→ Now, if you apply the greedy heuristic to the graph G' considering the nodes in the order of the numbers you just assigned, it is sure to find an optimal solution.

DisAdvantage:

- There are graphs, that make this heuristic as bad as you choose.
- Consider a graph with 2n nodes numbered 1 to 2n.
- When 'i' is odd, it is adjacent to all the even-numbered nodes. (except i+1)
- When 'i' is even, it is adjacent to all the odd-numbered nodes (except i-1)



← Bipartite graph.

$$\left.\begin{array}{l}1,3,5,7 \Rightarrow \text{red}\\ 2,4,6,8 \Rightarrow \text{blue}\end{array}\right\} k=2 \text{ (optimal)}$$

$$\text{In general} \Rightarrow \left.\begin{array}{l}1,3,\ldots,2n-1\\ 2,4,6,\ldots 2n\end{array}\right\} \text{will produce } k=2$$

$1, 2, 3, \ldots, 2n-1, 2n \Rightarrow k = n$ colours.

→ To protect against major errors, run the heuristic several times on the same graph, with different randomly chosen ordering of nodes

→ $\forall$ colour ($\forall$ node ($\forall$ neighbour))) $\Rightarrow O(n^3)$.

The Travelling Salesperson

→ NP-hard (known algorithms are impractical for large instances)

→ 4461 towns {IP, Cutting-plane techniques}

→ Problem can be represented as a complete undirected graph with n nodes.

→ to find the shortest Hamiltonian cycle in the given graph

6 towns:

From	To: 2	3	4	5	6
1	3	10	11	7	25
2		8	12	9	26
3			9	4	20
4				5	15
5					18

→ optimal tour length → 58

$1 \to 2 \to 3 \to 6 \to 4 \to 5$ ← TSP tour

Greedy heuristic:

- Start at an arbitrary node and then choose at each step to visit the nearest remaining unvisited node.

→ Start at node 1 → 2 → 3 → 5 → 4 → 6 → 1

Total cost → 60

Remark: For this example, the greedy solution is not far wrong from the optimal solution but can be catastrophic in extreme cases!!!