Q1)

Gini Index:

Gini index computes the degree of probability of a specific variable that is wrongly being classified when chosen randomly and a variation of gini coefficient. It works on categorical variables, provides outcomes either be “successful” or “failure” and hence conducts binary splitting only.

Information Gain:

Information gain is used for determining the best features/attributes that render maximum information about a class. It follows the concept of entropy while aiming at decreasing the level of entropy, beginning from the root node to the leaf nodes.

Information gain computes the difference between entropy before and after split and specifies the impurity in class elements.

Gini Index V/s Information Gain:

While Gini Index performs binary splitting in order to determine the efficiency of the split, Information Gain is robust. This is because of the fact that the algorithm itself experiments with the data in a greedy fashion in-order to land at the best possible split using the concept of entropy.

Even though Information gain is computationally heavier compared to Gini Index, we can be sure that it enables us to get the best possible split. However, if the data consists of large number of partitions, Information Gain will take lots of time to reach optimality. In such cases Gini Index could be preferred.

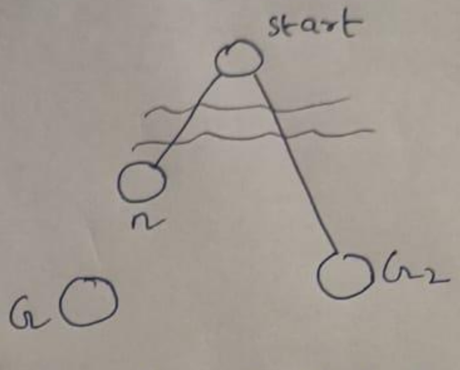
Example: If we have a dataset of 1000 classes, using information gain will take much time as compared to Gini index.

Q2)

Statement is true

- A heuristic is admissible if h(n)<=h*(n)
- Where h*(n) is the true cost to the goal→ h(n) never overestimates the cost from node to goal state.
- A* is guaranteed to always return the shortest path. All other algorithms return the optimal path must expand all other nodes whereas A* doesn't

Since we are using an admissible heuristic h(n) never overestimates the actual cost from node n to goal. Since A* heuristic is calculated in such a way that it adds both the cost to that node and cost from that node to Goal state ,A* is always guaranteed to return the shortest path and doesn’t expand to an useless node



n → unexpanded node that is on shortest path to Goal (G)

G_2 → Sub optimal goal G_2

$h(G_2) = 0 \Rightarrow f(G_2) = g(G_2)$

As G_2 is not optimal, $g(G_2) > g(G)$

as $h(G) = 0 \rightarrow f(G) = g(G)$

$\therefore f(G_2) > f(G)$

as n is unexpanded node, $f(n) = \underbrace{g(n) + h(n)}_{\leq f(G)}$

$f(n) \leq f(G)$ Since h is admissible

This implies $f(G_2) > f(n)$

A^* will never select G_2 for expansion

$\therefore$ A useless node will never be selected by A^*

Q3)

Given Knowledge base:

Ancestor (Mother(x), x)

Ancestor (x,y) ^ Ancestor (y,z) ==> Ancestor (x,z)

Clearly, we can see that the resolution is complete

We need to prove the following:

~ Ancestor (John, John)

The resolution cannot prove the above statement despite being complete or finished. A careful analysis will show that the statement to be proven doesn't follow the knowledge base.

Reason: There is nothing in the knowledge base that dismisses the possibility of everything being the ancestor of everything else.