Hill Climbing is heuristic search used for mathematical optimization problems in the field of Artificial Intelligence.

Features of Hill Climbing:

1. **Variant of generate and test algorithm :** It is a variant of generate and test algorithm. The generate and test algorithm is as follows :

    1. Generate a possible solutions.
    2. Test to see if this is the expected solution.
    3. If the solution has been found quit else go to step 1.

2. **Uses the Greedy approach** At any point in state space, the search moves in that direction only which optimizes the cost of function with the hope of finding the optimal solution at the end.

Types of Hill Climbing:

- **Simple Hill climbing :** It examines the neighboring nodes one by one and selects the first neighboring node which optimizes the current cost as next node.

    Algorithm for Simple Hill climbing :

    1. Evaluate the initial state. If it is a goal state then stop and return success. Otherwise, make initial state as current state.
    2. Loop until the solution state is found or there are no new operators present which can be applied to current state.
       a) Select a state that has not been yet applied to the current state and apply it to produce a new state.

       b) Perform these to evaluate new state
          i. If the current state is a goal state, then stop and return success.
          ii. If it is better than the current state, then make it current state and proceed further.
          Iii. If it is not better than the current state, then continue in the loop until a solution is found.

    3. Exit.

- **Steepest-Ascent Hill climbing :** It first examines all the neighboring nodes and then selects the node closest to the solution state as next node.

    Algorithm for Steepest-Ascent Hill climbing :

    1. Evaluate the initial state. If it is goal state then exit else make the current state as initial state.
    2. Repeat these steps until a solution is found or current state does not change.
        - Let ‘target’ be a state such that any successor of the current state will be better than it;

- or each operator that applies to the current state.
  a. apply the new operator and create a new state
  b. evaluate the new state
  c. if this state is goal state then quit else compare with ‘target’
  d. if this state is better than ‘target’, set this state as ‘target’
  e. if target is better than current state set current state to Target.

3. Exit

- **Stochastic hill climbing :** It does not examine all the neighboring nodes before deciding which node to select .It just selects a neighboring node at random, and decides (based on the amount of improvement in that neighbor) whether to move to that neighbor or to examine another.