**Implementation of backtracking algorithms**

**Hamiltonian circuit problem**

**Aim:**

To implement the Hamiltonian circuit problem using backtracking.

**Problem description:**

The Hamiltonian circuit problem involves finding a path in a given graph where all vertices are visited exactly once, and the path starts and ends at the same vertex.

**Algorithm:**

* + Start with an empty path and an empty set of visited vertices.
  + Choose a starting vertex arbitrarily and add it to the path and the set of visited vertices.
  + If all vertices are visited and the current vertex has an edge to the starting vertex, then we have found a Hamiltonian circuit. Print the path and terminate the algorithm.
  + If the current vertex has unvisited neighbors, choose an unvisited neighbor and add it to the path and the set of visited vertices.
  + Recursively call the algorithm with the new current vertex.
  + If the recursive call returns false, remove the last vertex from the path and the set of visited vertices.
  + Repeat steps 4-6 for all unvisited neighbors of the current vertex.
  + If no unvisited neighbor leads to a Hamiltonian circuit, return false.

**Code:**

g={"A":["C","B","D"],"B":["A","C","D"],"C":["A","B"],"D":["A","B"]}

a=[]

def hamiltonian\_circuit(g,l):

    for i in g:

        if i==l[-1]:

            for j in g[i]:

                if j not in l:

                    if len(l)+1==len(g):

                        l.append(j)

                        print("Hamiltonian circuit has been found and it is",l)

                        quit()

                    else:

                        m=l.copy()

                        m.append(j)

                        hamiltonian\_circuit(g,m)

hamiltonian\_circuit(g,["A"])

**Output:**



**TIME COMPLEXITY:**

* The Hamiltonian Circuit problem is known to be NP-complete, which means that there is no known polynomial-time algorithm to solve it for all instances. As a result, the time complexity of finding a Hamiltonian Circuit in a graph is exponential.
* The Hamiltonian Circuit problem is computationally challenging, and finding an optimal solution for all instances is believed to be intractable in terms of time complexity.

**ALGORITHM ANALYSIS:**

The brute-force approach to solve the Hamiltonian Circuit problem involves

generating all possible permutations of the vertices and checking each

permutation to see if it forms a Hamiltonian Circuit. The number of

permutations is factorial, which grows very quickly with the number of vertices.

Hence, the time complexity of the brute-force approach is O(n!), where n is the

number of vertices in the graph.

There are also more efficient algorithms and heuristics for solving the

Hamiltonian Circuit problem, such as backtracking algorithms, dynamic

programming, and branch-and-bound techniques. These algorithms can

improve the average-case performance for certain types of graphs or provide

approximate solutions. However, their worst-case time complexity remains

exponential.

**Result:**

Thus, Hamiltonian circuit has been solved using backtracking successfully.