Hamiltonian Cycles

Unit-IV - Backtracking

Habul's Pizza Store has received order from 5 customers: Anil, Bimal, Chetan, Dipak and Ekta. Habul needs to deliver a pizza to each of their houses, and then come back to the Pizza Store.

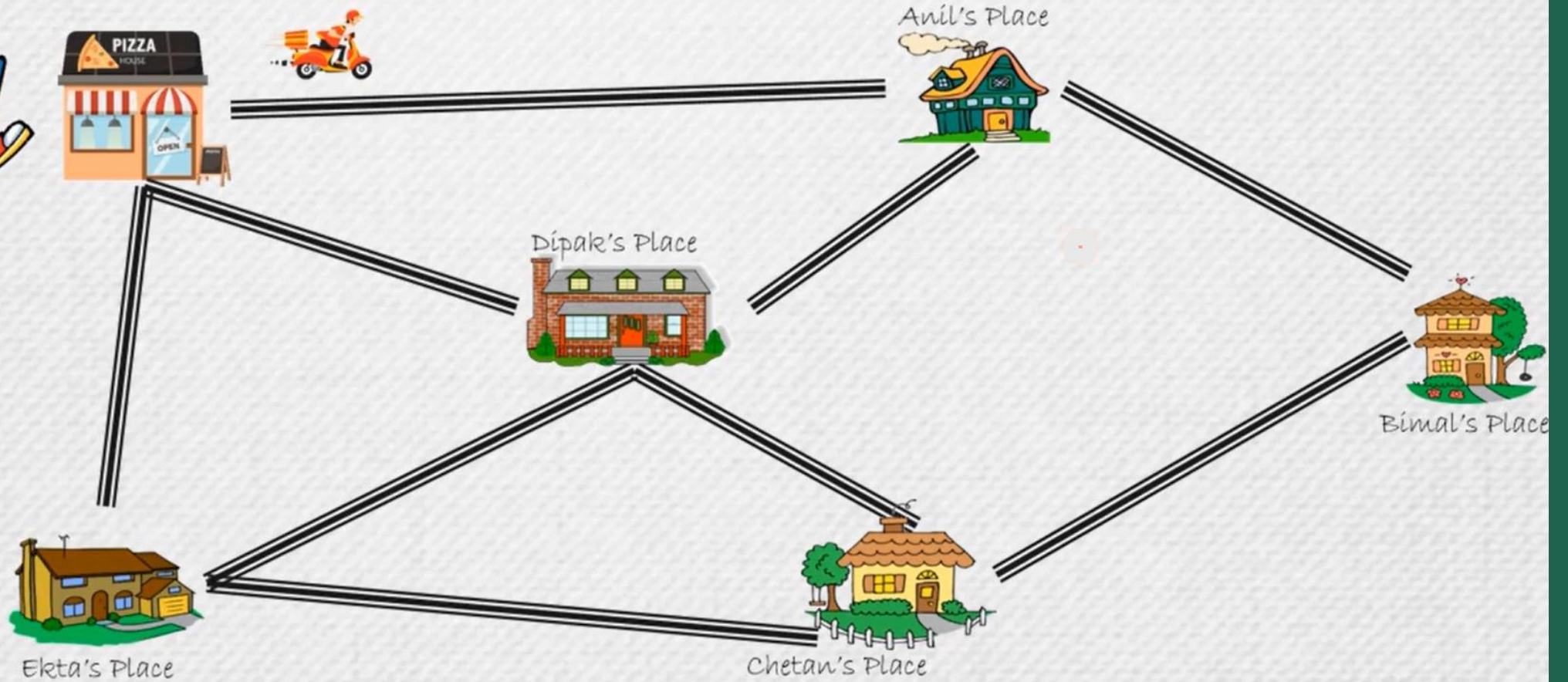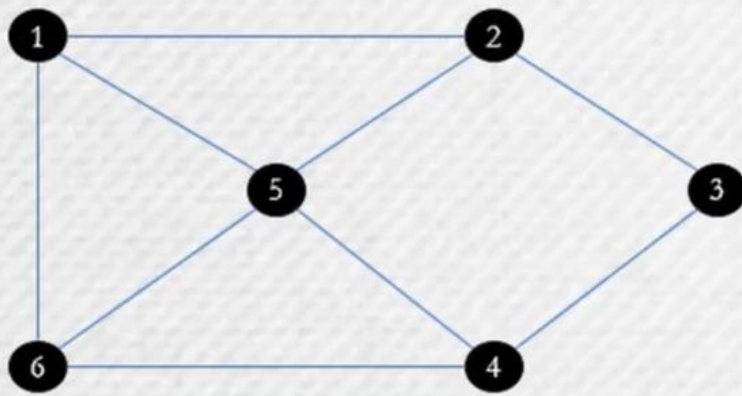Anil's Place

Dipak's Place

Bimal's Plac

Ekta's Place

Chetan's Place

Habul's Pizza Store has received order from 5 customers: Anil, Bimal, Chetan, Dipak and Ekta. Habul needs to deliver a pizza to each of their houses, and then come back to the Pizza Store. In how many ways can Habul do this?
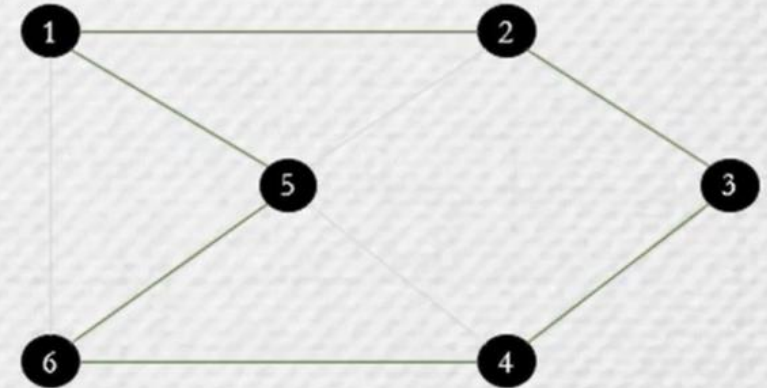
The problem can be rephrased as:
In how many ways can Habul start from vertex
1, then visit every other vertex of the graph
exactly **once** and come back to **1** ?

We need to find Hamiltonian Cycles:

A Hamiltonian cycle in a graph $G(V, E)$ is a round trip path in
G that starts from a given vertex $v \in V$, then visits every other
vertex of G exactly once and returns to the starting vertex v.

Solution using Backtracking

```
1   Start from a given vertex, say ❶
2   while there is an unvisited adjacent vertex
3       move to that vertex
4   if all the vertices are visited
5       if there is an edge from the last vertex to ❶
6           We have found a Hamiltonian Cycle
7       else Dead End go to step 9
8   else Dead End go to step 9
9   backtrack to the previous vertex
10  go to step 2
```
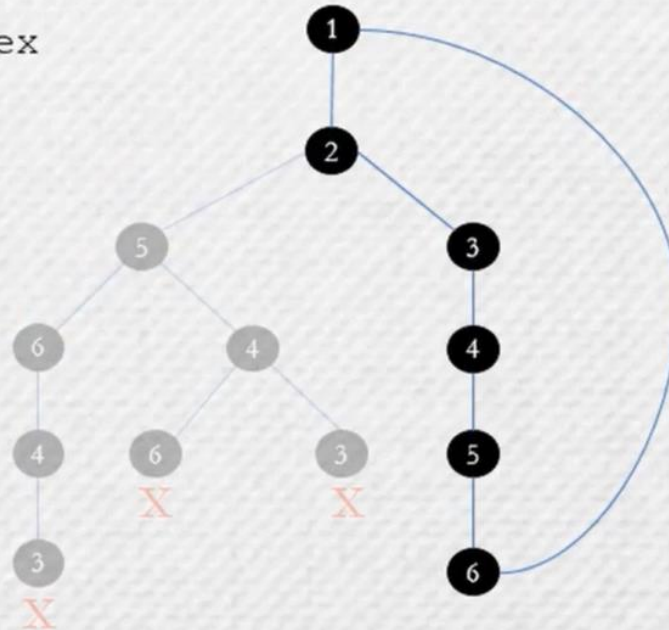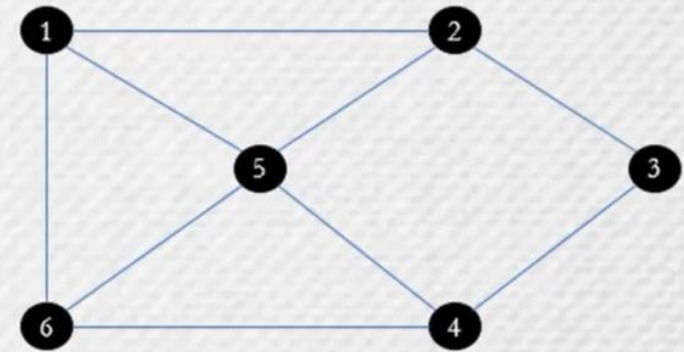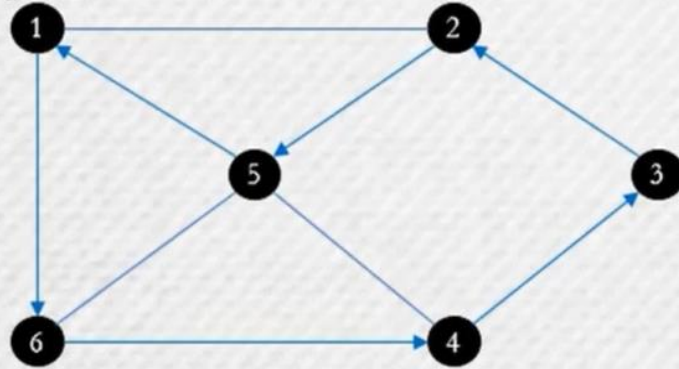
Let G[1 ... n][1 ... n] be the adjacency matrix representing the given graph.

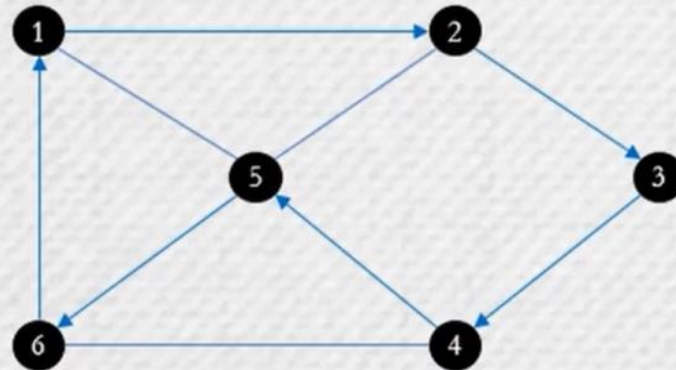We will present our solution in an one-dimensional array hc[1 ... n].
hc[1] is the first vertex (i.e. 1) in the Hamiltonian Cycle.
hc[2] is the second vertex in the Hamiltonian Cycle, and so on...
In general, hc[i] represents the ith vertex in the Hamiltonian cycle.

This Hamiltonian Cycle

	1	2	3	4	5	6
hc	1	6	4	3	2	5

Is represented by this array

This Hamiltonian Cycle

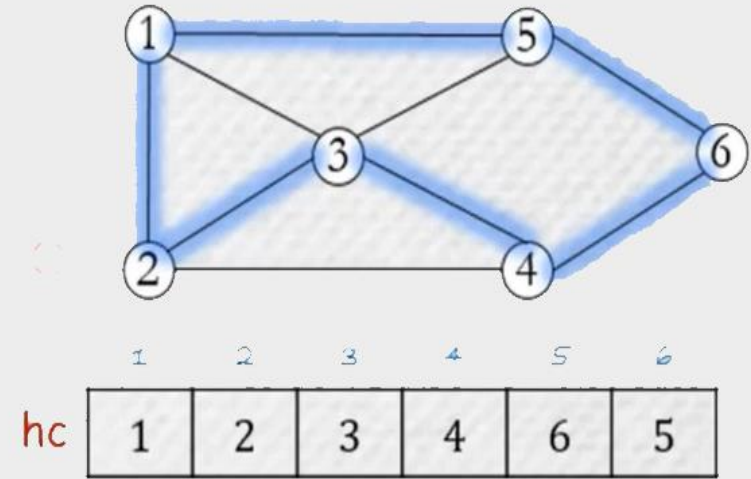|	1	2	3	4	5	6
hc	1	2	3	4	5	6

Is represented by this array

```
Algorithm chooseVertex(k)

1    do

2        hc[k] = (hc[k] + 1) mod (n + 1)

3        if (hc[k] == 0) return

4        if (G[hc[k - 1]][hc[k]] ≠ 0)

5            for (j = 1 to k - 1)

6                if (hc[k] == hc[j]) break

7            if (j == k)

8                if (k < n OR (k == n AND G[hc[n]][hc[1]] ≠ 0)) return

9    while (true)


Algorithm hamiltonianCycles(k)

1    do

2        chooseVertex(k)

3        if (hc[k] == 0) return

4        if (k == n) displayHamiltonianCycles()

5        else hamiltonianCycles(k + 1)

9    while (true)
```


hc

1	2	3	4	5	6
1	2	3	4	6	5

hamiltonianCycles(6)

hamiltonianCycles(5)

hamiltonianCycles(4)

hamiltonianCycles(3)

hamiltonianCycles(2)

Stack