Moscow Institute of Physics and Technology

HAMILTONIAN CYCLE

Final project
by Nguyen Dai Hiep

Course: **NP-Problem Practice**
Group: **ФПМИ - Б05-041**

Date of exam: **May 23, 2022**

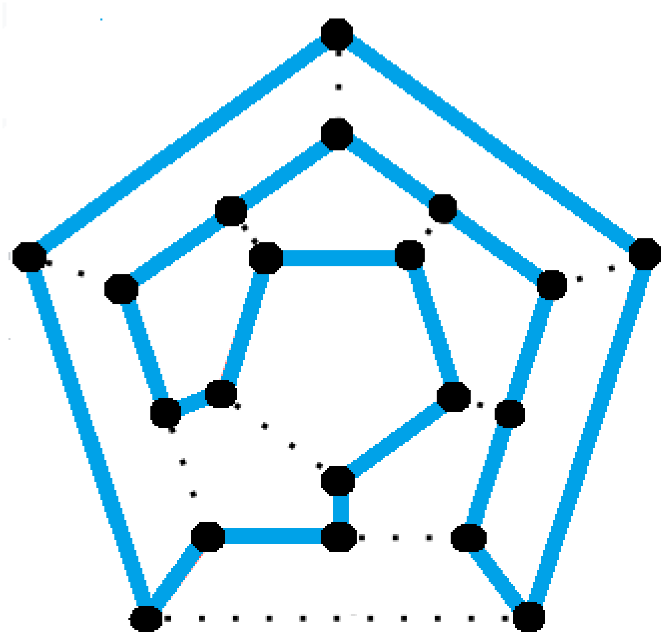
WHAT IS HAMILTONIAN CYCLE PROBLEM?

A Hamiltonian cycle is a closed loop on a graph where every node (vertex) is visited exactly once.

A loop is just an edge that joins a node to itself; so a Hamiltonian cycle is a path traveling from a point back to itself, visiting every node in route.

The Hamiltonian cycle problem is the problem of determining whether a Hamiltonian cycle exists in a given graph (whether directed or undirected).

Example:



ALGORITHMS SOLVING HAMILTONIAN CYCLE PROBLEM

1. **BRUTE FORCE**
2. **BELLMAN-HELD-KARP ALGORITHM**
3. **INCLUSION – EXCLUSION PRINCIPLE (ANDREAS BJORKLUND)**

Problem: Given a graph (Number of nodes and Adjacency matrix of the graph). Return `**True**` if the graph has Hamiltonian cycle, otherwise return `**False**`.

1. Brute Force

This is a way which tests all possible sequence path.

Create the list of all permutation of n-nodes. Each permutation represents for a path.
-> The next problem is check whether the path is valid.
- For each path:
    * Travel the path, if a edge of path doesn't exist, then the path isn't valid. If all of edges exist, the path is valid.

Return true if there are at least a valid path, otherwise return false.

<u>Time Complexity</u>: O(n!)

2. Bellman-Held-Karp

Like Traveling Salesman problem but without the cost of each pair.

By dynamic programming, we have to check that, for the s-nodes subset (subset includes s nodes), which has k is the last node of path, is valid. If the subset {S \ i} (subset S without i), with j is the last node, is valid and there is an edge between i and j, then the subset {S} with i is the last node, is also valid. Let's choose the fixed node is node 1. In initiation, check if C({i}, i) (is Boolean 2-dimension array: subset {i}, and i the last node) is valid when g(1, i) == 1 or g(i, 1) == 1.

To represent all subset of (n-1)-nodes, we use bit-masking method, in example: number of node is 6, $000101_2 = 5_{10}$ means the path {1,2,4}.

Dynamic programming formula: (Pseudo Code)

```
For s = 2 to n – 1
   For each S
     For each i
          For each j
               C(S, i) = C(S \ i, j) and (g(i,j) or g(j,i))
                where:        - S is a subset with s-nodes.
                        - i is a node in S.
                        - j is other node in S.
                               - g is adjacency matrix.
```

After that, with the subset S = {2,3,…,n} and ∀ i ∈ S, check C(S, i) and (g(1,i) or g(i,1)) is valid. If there is at least an i-node, which the expression is valid, then return true, otherwise return false.

<u>Time Complexity</u>: O($n^2 2^n$)

3. Inclusion – Exclusion principle

It is developed by **Bellman-Held-Karp algorithm**.

It is originally an answer to question of can we reduce to **$O(c^n)$** where **$c < 2$**. Bjorklund has written article about it and according to this article we can reduce **$O(1.657^n)$** by **using inclusion-exclusion approach** of possible set and calculate the determinant of adjacency matrix.

It divides the vertices into two groups, one of them labels, then calculates the possibility of Hamilton cycle using determinant.

It uses Monte-Carlo simulation to calculate the Hamilton cycle.

<u>Time Complexity</u>: $O(1{,}657^n)$.