제목 : 해밀턴 회로 기술보고서
학과 : 컴퓨터학부
학번 : 20172605
이름 : 김도현
과목 : 컴퓨터수학
담당교수 : 길아라 교수님
작성일자 : 2017.6.4

목차

1. 서론

1.1 분야

Hamiltonian Circuit Problem은 1857년 아일랜드 수학자 Sir William Rowan Hamilton에 의해 발명된 Icosian puzzle이라는 게임에서 유래했다. Hamiltonian Circuit Problem은 어떠한 그래프가 주어졌을 때, “그래프의 모든 정점들을 정확히 한번씩만 포함하는 순환(Circuit)이 존재하는가?”를 판별하는 문제이다.

1.2 동기

해밀턴 회로 판별문제는 대표적인 NP-완전(다항시간 알고리즘을 찾지 못한)문제이다. 이 문제를 다항시간 내에 풀 수 있다면 필즈상, 튜링상은 물론이고 돈과 명예도 얻을 수 있다고 한다. 그래서 흥미를 가지게 되었고, 지금까지 공부한 내용을 통해 -여전히 다항시간은 아니지만- 풀어보았다.

1.3 목적

주어진 그래프가 해밀턴 회로를 갖는지 판별하는 자동화된 프로그램 작성

1.4 연구방법

인터넷 검색, Computational Thinking, 프로그래밍, 디버깅

2. 관련 연구

“Fault-free Hamiltonian cycles in crossed cubes with conditional link faults”(Hung Hao-Shun 외 2 명, Elsevier / Information Sciences)

“A fault-free Hamiltonian cycle passing through prescribed edges in a hypercube with faulty edges”(Wang Wen-Qing 외 1 명, Elsevier / Information Processing Letters)

둘 다 SCI급 학회에 실린 논문인데, Fault Tree에 관련하여 Hamiltonian Cycle을 푸는 것 같다. 특히 2번째 내용은 4차원 정육면체(hypercube)에서 Hamiltonian Cycle을 연구하는 내용이라서 신기했다. Fault Tree가 정확히 무엇을 하는 것인지는 모르지만, 최근에는 그런 내용이 주로 연구된다는 것은 알겠다.

3. 본론-설계

3.1 미션에 성공할 수 있는 지도의 일반적인 조건(도시와 연결 도로와의 관계)은 무엇인지 일반식으로 나타내어라.

오일러 회로와는 달리 해밀턴 회로는 판별을 위한 다항시간 알고리즘이 알려져 있지 않다. 그러나 가능한 모든 경우를 시도해보면 판별이 가능하다(=완전탐색 알고리즘). 따라서, 미션에 성공할 수 있는 지도의 일반적인 조건은 없다고 할 수 있지만, 가능한 모든 경우를 시도해보면 Hamiltonian Circuit인지 알 수 있다.

3.2 3.1의 일반 조건을 토대로 (n X n) 행렬형태로 주어지는 임의의 n개의 도시 지도에 대하여 미션 성공 여부를 판단하는 프로그램을 작성하고 판단과정을 출력하여라. (단, 도시와 도시를 연결하는 모든 도로의 길이는 같다고 가정한다) (Hint: hamiltonian circuit problem)

3.3 문제분석

3.3.1 용어 정의

해밀턴 경로 : 그래프의 모든 정점을 한번씩만 지나는 길

해밀턴 회로 : 그래프의 모든 정점을 한번씩만 지나서 다시 시작지점으로 돌아오는 길

∴ 해밀턴 회로는, 시작지점을 2번까지 방문가능하고, 시작지점과 종료지점이 같다는 조건이 추가된 해밀턴 경로라고 할 수 있다.

3.3.2 패턴 분석

손으로 몇 개의 문제를 풀어본 결과, 다음과 같은 문제 분할과 재귀적 패턴을 찾아내었다.

3.3.2.1 해밀턴 회로 판별문제의 분할

i) 정점 하나로 구성된 그래프의 경우

자명하게 해밀턴 순환을 갖는다.

<자명한 증명> 이 그래프에 존재하는 단 하나의 정점에 방문하면, 모든 정점을 한번씩만 방문한 것이고, 시작정점과 종료정점이 같으므로, 해밀턴 회로의 정의에 의해 해밀턴 순환을 갖는다.

ii) 아닌 경우

해밀턴 회로 판별의 시작정점을 a라 하자. 그러면 a와 연결된 정점에서 시작하여 a에서 종료하는 해밀턴 경로가 하나라도 존재한다면, 해밀턴 회로가 존재한다.

<직접 증명> a와 연결된 정점에서 시작하여 a에서 종료하는 해밀턴 경로가 있다고 하자. 그 해밀턴 경로의 시작정점(a와 연결된 정점)앞에 a를 추가하면 시작정점과 종료정점이 둘 다 a로 같으므로 해밀턴 회로가 된다.

3.3.2.2 해밀턴 경로 판별문제의 재귀적 패턴

기본 단계) 정점 하나로 구성된 그래프는 자명하게 해밀턴 경로를 갖는다.

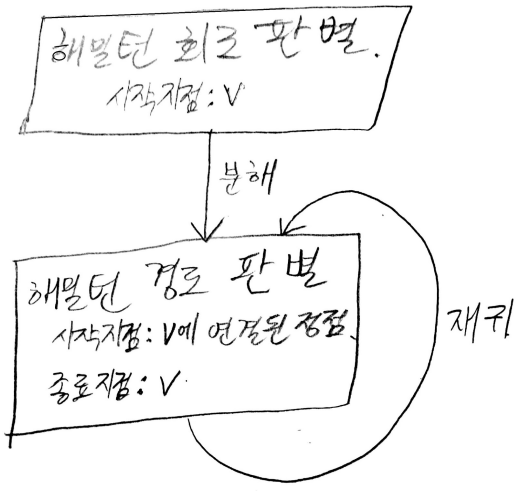
<자명한 증명>이 그래프에 존재하는 단 하나의 정점에 방문하면, 모든 정점을 한번씩만 방문한 것이므로 해밀턴 경로의 정의에 의해 해밀턴 경로를 갖는다.

재귀적 단계) 해밀턴 경로 판별의 시작정점을 a, 종료정점을 b라 하자. 정점 a가 제거된 그래프에 대해, a와 연결되있던 정점에서 시작하여 b에서 종료하는 해밀턴 경로가 하나라도 존재한다면, 원래의 그래프에도 a에서 시작하여 b에서 종료하는 해밀턴 경로가 존재한다.

<직접 증명> a가 제거된 그래프에 대해, a와 연결되있던 정점에서 시작하여 b에서 종료하는 해밀턴 경로가 존재한다고 하자. 그 경로의 시작정점(a에 연결되이던 정점) 앞에 a를 추가하면, 원래의 그래프에 대해 모든 정점을 한번씩만 지나게 되므로 해밀턴 경로가 된다. 그리고 그 해밀턴 경로는 a에서 시작하여 b에서 종료한다.

3.4 디자인

해밀턴 회로는 특수한 조건이 적용된 해밀턴 경로라는 것에 착안하여, 해밀턴 회로 판별기능에서 해밀턴 경로 판별기능을 분리했다. 해밀턴 경로 판별기능은 문제분석에서 찾아낸 재귀적 패턴을 활용하여 재귀적으로 디자인하였다.



▲ 해밀턴 회로 판별함수와 해밀턴 경로 판별함수 사이의 관계

해밀턴 경로 판별시 그래프에서 정점을 지우고, 경로가 없다면 다음 탐색을 위해 정점을 다시 복구해야 한다. 이러한 동작방식을 위해 백트래킹 기법을 사용한다. 백트래킹은 탐색실패 시 이전상태로 되돌아가 다시 검색해나가는 기법으로, 대표적인 백트래킹 문제로는 N-Queen문제 등이 있다.

3.5 알고리즘 의사코드 설계

```
//해밀턴 회로 판별
procedure IsHamiltonianCircuitExist(g:graph, from:vertex)
{
        if IsOnlyOneVertexLeft(g) then
                return true

        for  v∈ConnectedVertices(g, from)
        {
```

```
                if IsHamiltonianPathExist(g, v, from) then
                        return true
        }
        return false
}

//해밀턴 경로 판별
procedure IsHamiltonianPathExist(g:graph, from:vertex, to:vertex)
{
        if from==to && IsOnlyOneVertexLeft(g) then
                        return true

        vertices = ConnectedVertices(g, from)

        for v ∈ vertices
                RemoveEdge(g, from, v)
        RemoveVertex(g, from)

        for v ∈ vertices
                if IsHamiltonianPathExist(g, v, to) then
                        return true

        AddVertex(from)
        for v ∈ vertices
                AddEdge(g, from, v)

        return false
}
```

4. 구현 또는 모의실험

사용한 기기 : Hansung H57 DGA7700
시스템 구성 : Windows 10 + Visual Studio 2015 Community
※ Windows Server 2008 R2 Enterprise(N Cloud) + Cygwin 및 Ubuntu + gcc 환경에서도 정상동작함을 확인함.

4.1 입력 형식

정점의 개수 : vn (unsigned int 형)
간선의 개수 : en (unsigned int 형)
간선 0 : 정점이름 정점이름
…
간선 en-1 : 정점이름 정점이름

4.2 주의사항

-정점의 개수는 반드시 1개 이상 100개 이하
-간선의 개수는 반드시 0개 이상 100*100개 이하
-정점이름은 0부터 vn-1까지이다 (1부터 시작하지 않는다.)

4.3 IsHamiltonCircuitExist()함수의 시간복잡도

n개의 정점을 갖는 그래프에 대해 Theta(n!)의 시간복잡도를 가진다. 왜냐하면 완전그래프에서 1을 제외한 모든 노드와 0 사이의 간선을 지우면 최악의 경우가 발생하게 되는데; 해밀턴 회로가 아님이 판별되기까지 처음에

1가지, 두번째에 n-1가지, 세번째에 n-2가지, 그 뒤로는 모두 n-2가지가 있다. 곱셈법칙에 의해 총 (n-1)(n-2)^(n-2)번의 IsHamiltonianPathExist()가 호출되므로 최악의 시간복잡도가 Theta(n!)이다.

4.4 예제 입력 및 출력

입력:

```
20
30
0 1
1 2
2 3
3 4
4 0
0 5
1 6
2 7
3 8
4 9
5 11
11 6
6 12
12 7
7 13
13 8
8 14
14 9
9 10
10 5
10 15
11 16
12 17
13 18
14 19
15 16
16 17
17 18
18 19
19 15
```

출력:

```
도시0 방문
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문
도시3 방문
도시4 방문
도시0 방문
도시0 방문 취소(backtrack)
도시9 방문
도시10 방문
도시5 방문
도시0 방문
도시0 방문 취소(backtrack)
```

도시11 방문
도시6 방문
도시12 방문
도시7 방문
도시13 방문
도시8 방문
도시14 방문
도시19 방문
도시15 방문
도시16 방문
도시17 방문
도시18 방문
도시18 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시18 방문
도시17 방문
도시16 방문
도시15 방문
도시15 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시18 방문
도시17 방문
도시16 방문
도시15 방문
도시19 방문
도시14 방문
도시8 방문
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시15 방문
도시16 방문
도시17 방문
도시17 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)

도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시17 방문
도시16 방문
도시15 방문
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시18 방문
도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시18 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시18 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문
도시19 방문
도시15 방문
도시16 방문
도시16 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시13 방문 취소(backtrack)

도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시15 방문
도시16 방문
도시16 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시16 방문
도시15 방문
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시17 방문
도시18 방문
도시18 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시18 방문
도시17 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문
도시7 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시18 방문
도시13 방문
도시7 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시17 방문
도시17 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문

도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시17 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문
도시13 방문
도시8 방문
도시14 방문
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시15 방문 취소(backtrack)
도시17 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문
도시13 방문
도시8 방문
도시14 방문
도시19 방문
도시15 방문
도시15 방문 취소(backtrack)
도시18 방문
도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시18 방문
도시19 방문
도시14 방문
도시8 방문
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시15 방문
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시18 방문
도시13 방문
도시7 방문

도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문
도시19 방문
도시15 방문
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시12 방문
도시6 방문
도시6 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시15 방문
도시15 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시16 방문 취소(backtrack)
도시11 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시15 방문
도시16 방문
도시11 방문
도시5 방문
도시0 방문
도시0 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시12 방문
도시7 방문
도시13 방문
도시8 방문
도시14 방문
도시19 방문
도시18 방문
도시17 방문
도시17 방문 취소(backtrack)

도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시18 방문
도시17 방문
도시17 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시17 방문
도시18 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문
도시19 방문
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시7 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시17 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시11 방문 취소(backtrack)
도시17 방문
도시12 방문
도시6 방문
도시11 방문
도시5 방문
도시0 방문
도시0 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시11 방문 취소(backtrack)

도시6 방문 취소(backtrack)
도시7 방문
도시13 방문
도시8 방문
도시14 방문
도시19 방문
도시18 방문
도시18 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시18 방문
도시19 방문
도시14 방문
도시8 방문
도시8 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시19 방문 취소(backtrack)
도시18 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시18 방문
도시13 방문
도시7 방문
도시12 방문
도시6 방문
도시11 방문
도시5 방문
도시0 방문
도시0 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시11 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시12 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시8 방문
도시14 방문
도시19 방문
도시19 방문 취소(backtrack)
도시14 방문 취소(backtrack)
도시8 방문 취소(backtrack)
도시13 방문 취소(backtrack)
도시19 방문
도시14 방문
도시8 방문
도시13 방문
도시7 방문
도시12 방문
도시6 방문
도시11 방문
도시5 방문

도시0 방문
해밀턴 회로가 존재합니다
회로 : 0 5 11 6 12 7 13 8 14 19 18 17 16 15 10 9 4 3 2 1 0

입력:
6
10
0 1
1 2
2 3
3 4
4 0
1 4
1 5
4 5
2 5
3 5
출력:
도시0 방문
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문
도시3 방문
도시4 방문
도시0 방문
도시0 방문 취소(backtrack)
도시5 방문
도시5 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시5 방문
도시4 방문
도시0 방문
해밀턴 회로가 존재합니다
회로 : 0 4 5 3 2 1 0

입력:
8
13
0 1
1 3
3 2
2 0
0 3
1 2
3 4
4 5
5 7
7 6
6 4
4 7
5 6

출력:
도시0 방문
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문
도시0 방문
도시0 방문 취소(backtrack)
도시3 방문
도시0 방문
도시0 방문 취소(backtrack)
도시4 방문
도시5 방문
도시6 방문
도시7 방문
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시7 방문
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시7 방문
도시5 방문
도시5 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시5 방문
도시6 방문
도시6 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시5 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시3 방문 취소(backtrack)
도시2 방문 취소(backtrack)
도시3 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문 취소(backtrack)

도시4 방문
도시5 방문
도시6 방문
도시7 방문
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시7 방문
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시7 방문
도시5 방문
도시5 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시5 방문
도시6 방문
도시6 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시5 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시3 방문 취소(backtrack)
도시1 방문 취소(backtrack)
도시2 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시3 방문
도시0 방문
도시0 방문 취소(backtrack)
도시4 방문
도시5 방문
도시6 방문
도시7 방문
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시6 방문
도시6 방문 취소(backtrack)

도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시7 방문
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시7 방문
도시5 방문
도시5 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시5 방문
도시6 방문
도시6 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시5 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시3 방문 취소(backtrack)
도시1 방문 취소(backtrack)
도시3 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문 취소(backtrack)
도시4 방문
도시5 방문
도시6 방문
도시7 방문
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시7 방문
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시7 방문
도시5 방문
도시5 방문 취소(backtrack)
도시7 방문 취소(backtrack)

도시6 방문 취소(backtrack)
도시7 방문
도시5 방문
도시6 방문
도시6 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시5 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시3 방문 취소(backtrack)
도시2 방문 취소(backtrack)
도시3 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문
도시0 방문
도시0 방문 취소(backtrack)
도시2 방문 취소(backtrack)
도시1 방문 취소(backtrack)
도시2 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문
도시0 방문
도시0 방문 취소(backtrack)
도시1 방문 취소(backtrack)
도시2 방문 취소(backtrack)
도시4 방문
도시5 방문
도시6 방문
도시7 방문
도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시6 방문
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시7 방문
도시7 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시7 방문
도시5 방문
도시5 방문 취소(backtrack)

도시7 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문
도시5 방문
도시6 방문
도시6 방문 취소(backtrack)
도시5 방문 취소(backtrack)
도시6 방문
도시5 방문
도시5 방문 취소(backtrack)
도시6 방문 취소(backtrack)
도시7 방문 취소(backtrack)
도시4 방문 취소(backtrack)
도시3 방문 취소(backtrack)
도시0 방문 취소(backtrack)
해밀턴 회로가 존재하지 않습니다

5. 결론, 향후 연구방향

Theta(n!)의 시간복잡도이므로 간단한 실생활정도에 이용할 수 있으나, 도로 설계같은 대규모 데이터에는 적합하지 않다. 구현한 알고리즘은 완전탐색 기법을 사용하고 있으므로, 조합탐색의 가지치기법을 통한 알고리즘 성능향상을 기대해볼 수 있다. 예를들어 그래프(C)의 경우, 완전탐색을 시행하기 전에 절단간선찾기를 시행함으로써 미리 제거할 수 있다. 절단간선찾기는 다항시간 알고리즘이 존재하기 때문에 절단간선이 없더라도 늘어나는 시간은 미미하므로 사용가치가 있다. 그 밖에 디랙의 정리, 오레의 정리 등의 조건들로 가지치기가 가능하다. Theta(1.657^n)의 시간복잡도를 갖는다는 몬테카를로 알고리즘도 공부해봐야겠다. 해밀턴회로가 NP-완전문제이기 때문에 이를 해결할 다항시간 알고리즘을 연구해볼 필요성도 있다.

6. 참고문헌

https://ko.wikipedia.org/wiki/%ED%95%B4%EB%B0%80%ED%84%B4_%EA%B2%BD%EB%A1%9C
https://namu.wiki/w/%ED%95%B4%EB%B0%80%ED%84%B4%20%ED%9A%8C%EB%A1%9C
https://ko.wikipedia.org/wiki/%ED%87%B4%EA%B0%81%EA%B2%80%EC%83%89