한국수학사학회지 제18권 제1호(2005년 2월), 103-110

그래프 이론의 역사적 배경과 그 컴퓨터 표현

안양대학교 컴퓨터학과 **김화준**
khj1@anyang.ac.kr

안양대학교 컴퓨터학과 **한수영**
syhan@anyang.ac.kr

스위스 수학자 오일러에 의하여 해결된 쾨니히스베르크의 다리문제에 대한 역사적 배경과 그 응용으로서 그래프의 컴퓨터 표현에 대하여 간단한 예를 통하여 행렬로 표현하였고, 오일러 회로에 의한 행렬 표현을 연구해보았다.

주제어: 그래프 이론, 인접행렬, 오일러 회로

0. 서론

그래프 이론의 역사적 배경과 응용으로서의 그 컴퓨터 표현에 대하여 조사・연구를 하였다. 그래프 이론은 실생활에서 일어나는 복잡한 문제를 쉽게 해결할 수 있게 도와주는 역할을 한다. 각각의 문제 상황에 알맞게 꼭지점과 변을 대응시켜서 그래프 상에서 문제를 해결한 후, 실생활 문제에 적용하여 그 타당성을 생각하는 것이다.

이 그래프 이론은 조직 문제, 사회구조 문제, 비행 항로, 최단거리 찾기, 정보 체계 문제, 자료 구조, 컴퓨터 구조, 알고리즘, 네트워크 등 무수히 많은 실용적인 문제에 응용되고 있다.

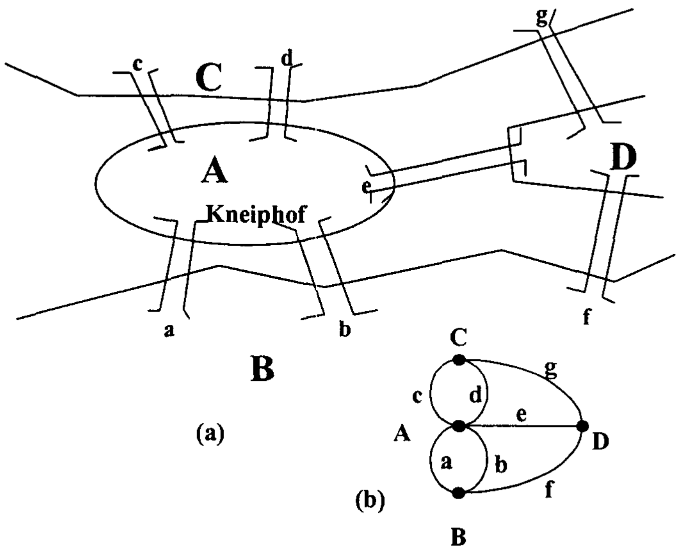
1. 역사적 배경

그래프 이론은 스위스 수학자 오일러(Leonhard Euler)에 의하여 1736년에 최초로 논문으로 발표되었다. 오일러 이전에는, 그래프 이론은 오락용 문제로 제기되고 있었다. 동 프러시아의 쾨니히스베르크(Koenigsberg, 지금의 칼리닌그라드)의 프레겔(Pregal) 강에는 서로 연결된 두 개의 섬이 있고, 이 섬들은 7개의 다리로 본토와 연결되어 있다. 문제는 각 다리를 정확히 한 번만 건너서 도시를 통과하여 걸어가는 계획이었다. 즉 7개의 다리를 모두 그러나 오직 한 번 건너면서 출발한 자리로 돌아오

는 방법이 있느냐는 것이었다. 많은 사람들은 경험적으로 이러한 방법이 존재하지 않을 것이라고 생각했으나 명쾌한 답을 구하지는 못하였다.

이 문제는 오일러의 관심을 끌게 되었고, 결론적으로 위 문제는 불가능하다는 사실을 증명하였다. 그 내용은 정점에 연결된 모서리의 수를 그 정점의 차수(degree)라 할 때, 위의 문제를 만족하는 해는 그래프의 모든 정점들이 짝수 차수를 가질 때만 가능하다는 것이었다. 참고로 아래에 이 문제의 그래프를 그려놓았다. 이 쾨니히스베르크의 다리 문제로부터 그가 위상수학을 창시한 것으로 받아들여진다[1].

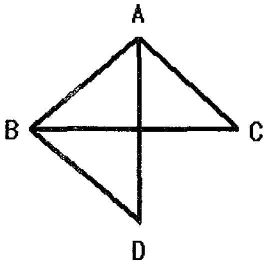
[그림 1] 쾨니히스베르크의 프레겔 강의 일부와 오일러의 그래프



2. 그래프의 컴퓨터 표현

우리는 정점들을 연결선으로 연결하여 그래프를 표현하는 것이 매우 편리함을 알고 있다. 그러나 그러한 표현은 그래프를 컴퓨터에 기억시키고자 할 때에는 부적당하다. 그러므로 이러한 경우에 사용되는 표현이 대수적 표현 방법인 그래프의 행렬 표현이다[5]. 그러면 고등학교 이산수학 교과서에 나와있는 간단한 예를 보자. 아래 그래프는 네 도시 사이의 개통된 고속도로를 나타낸 것이다[2]. 이 그래프에서 도시와 도시 사이의 관계를 행렬로 나타내어보자.

[그림 2] 고속도로



(1) 인접행렬(adjacency matrix)에 의한 표현

[표 3] 인접행렬에 의한 표현

	A	B	C	D
A	0	1	1	1
B	1	0	1	1
C	1	1	0	0
D	1	1	0	0

각 행렬의 성분은 두 도시 사이에 개통되어 있는 길(path)의 수를 의미한다. 즉 몇 개의 모서리들이 정점들을 연결하는가를 보인다. 인접행렬을 A라 하면 행렬 A^2은 한 도시를 경유해 다른 도시로 가는 방법의 수를 나타낸다. 다시 말해서 $A^2 = B = (b_{ij})$라면, 다음과 같이 표시된다.

$$b_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj} \quad (\text{단, } A = a_{ij})$$

여기서 b_{ij}는 i에서 한 정점을 경유하여 j로 가는 길의 수를 의미한다. 위 행렬에서 \쪽 대각선 아래의 성분은 사용할 필요가 없는데, 그 이유는 A에서 B를 연결하는 모서리의 수가 B에서 A를 연결하는 모서리의 수와 같기 때문이다. 만약 주어진 그래프가 방향그래프(directed graph)라면, 전체 행렬을 사용하여 방향성 정보를 표현할 수 있다. 그래프가 단순(simple)하다는 것은 루프(loop)나 다중 모서리(multi edge)가 없다는 의미이다. 그래프가 단순하다면 인접행렬의 \쪽 대각선상의 성분은 모두 0이고, 대각선 위쪽에 있는 모든 수들은 0이거나 1이다. 이 인접행렬에 의한 표현은 가장 일반적인 그래프 표현 방법으로서, 편리하지만 모서리의 수가 커지면 컴퓨터의 기억용량을 낭비하는 단점이 있다.

(2) 모서리표(edge table)에 의한 표현

모서리 AB를 e_1, BC를 e_2, CA를 e_3, DA를 e_4, BD를 e_5라 하고 정점 A를 v_1, 정점 B를 v_2, 정점 C를 v_3, 정점 D를 v_4라 하면 다음과 같은 입사행렬(incidence matrix)을 얻을 수 있다.

[표 4-1] 입사행렬에 의한 표현

	v_1	v_2	v_3	v_4
e_1	1	1	0	0
e_2	0	1	1	0
e_3	1	0	1	0
e_4	1	0	0	1
e_5	0	1	0	1

(e_i, v_j)에서 1은 모서리 e_i가 정점 v_j로 입사되는 것을 나타낸다. 이 입사행렬의 함축된 형태를 모서리표라고 한다.

[표 4-2] 모서리표에 의한 표현

모서리	첫 번째 정점	두 번째 정점
1	1	2
2	2	3
3	1	3
4	1	4
5	2	4

예를 들어, 위 표에서 세 번째 줄은 모서리 e_3이 정점 v_1과 정점 v_3을 연결하고 있음을 나타낸다. 이와 같이 컴퓨터에서 그래프를 나타내는 가장 간단한 방법이 바로 모서리표에 의한 방법이다. 이것은 각 모서리를 나열하고 그것으로 들어가는 정점들을 나타낸다. 이 표에서 각 모서리의 끝점을 첫 번째 정점이라 부르고, 다른 정점을 두 번째 정점이라 부른다. 이 방법은 간단하지만, 모든 모서리를 다 찾아서 표현해야 하기 때문에 편리한 방법은 아니다. 만약 주어진 그래프가 방향그래프(directed graph)

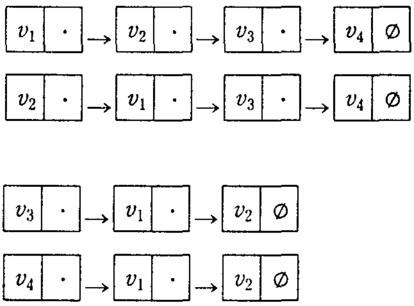
라면, 첫 번째 정점을 시작 정점으로 표시하고, 두 번째 정점을 끝 정점으로 표시하면 충분하다.

(3) 인접리스트(adjacency lists)에 의한 표현

그래프가 단순(simple) 그래프일 때는 컴퓨터의 기억 용량을 절약하기 위한 방법으로 인접리스트 방법을 사용한다. 이 방법은 각 정점에 대하여 그것에 인접한 정점들의 리스트들만 보관하는 방법이다.

위 고속도로 예를 보자. v_1은 v_2, v_3와 v_4에 인접하고, v_2는 v_1, v_3와 v_4에 인접하고, v_3는 v_1과 v_2에 인접하고, v_4는 v_1과 v_2에 인접하고 있다. 이 그래프는 정점표(vertex table)와 인접 리스트표(adjacency lists)의 2개의 구조로 컴퓨터에 저장된다. 정점표에 있는 각 항은 인접 리스트표에 있는 항으로의 포인터(pointer)를 갖는다. 인접리스트에 있는 각 항은 정점을 인식하고, 다음 항으로의 포인터를 포함한다. 널 포인터(null pointer)는 리스트의 마지막 항을 의미한다.

[그림 5] 그림 2의 그래프에 대한 인접리스트



[표 6] 인접 리스트에 의한 표현

[표 6-1] 정점표

정점의 번호	아이템 포인터
1	1
2	4
3	7
4	9

[표 6-2] 인접 리스트표

아이템 번호	정점의 번호	아이템 포인터
1	2	2
2	3	3
3	4	0
4	1	5
5	3	6
6	4	0
7	1	8
8	2	0
9	1	10
10	2	0

만약 주어진 그래프가 방향그래프(directed graph)라면, 두 개의 인접리스트에 의해서 나타낼 수 있다. 하나는 각 정점으로 들어오는 모서리(in pointer)에 대한 것이고, 다른 하나는 각 정점에서 나가는 모서리(out pointer)에 대한 것이다.

(4) 오일러 회로에 의한 행렬 표현

다음은 정점 A를 v_1, 정점 B를 v_2, 정점 C를 v_3, 정점 D를 v_4이라 하자. 위의 행렬 표현을 정점들의 연결 상태를 표현하기 위해서 (123142)에 의하여 표현하는 방법을 생각해보았다. 여기서 정점 3은 정점 2와 정점 1에 연결해 있음을 의미하고, 마지막 2는 정점 4가 정점 2와 연결되어있음을 의미한다.

이 방법은 오일러 회로를 표현하는 데 사용하는 방법인데 이 방법을 일반그래프로 확장해 생각해보아도 무리가 없었다. 위에서 언급한 오일러 회로는 연결된 그래프에서 정점은 여러 번 지날 수 있지만 모든 변을 오직 한 번씩만 지나는 회로를 의미한다. 만약 주어진 그래프가 방향그래프(directed graph)라면, 왼쪽 정점을 시작 정점으로 표시하고, 오른쪽 정점을 끝 정점으로 표시하면 가능하리라 여겨진다.

이 방법을 단순(simple)그래프가 아닌 경우로 확장해 생각해보자. [그림 1]의 쾨니히스베르크의 프레겔 강의 다리 문제는 정점 사이에 길의 수가 여러 개있으므로 단순그래프가 아니다. 먼저 인접행렬에 의하여 표시해보자.

[표 7] 그림 1의 인접행렬에 의한 표현

	A	B	C	D
A	0	2	2	1
B		0	0	1
C			0	1
D				0

그림 1의 쾨니히스베르크의 프레겔 강의 다리 문제를 이 방법으로 나타내면 (12(2)431(2)4)이다. 여기서 괄호 속의 수는 왼쪽 정점에서 오른쪽 정점으로 가는 길(walk)의 개수를 의미한다. 여기서 괄호 대신에 제곱의 형태를 취해도 좋을 듯 싶다. 표시하면 (12^2431^24)로 표시된다. 이 방법은 완전(complete)그래프일 때 효율적으로 적용되어지는 것으로 판단된다. 완전그래프는 서로 다른 정점 사이에 항상 모서리가 존재하는 그래프를 의미한다.

3. 결론

무수한 실용적인 문제에 응용되고 있는 그래프 이론의 역사적 배경과 그 컴퓨터 표현을 인접행렬, 모서리표, 인접리스트, 오일러 회로에 의한 행렬표현에 의해 나타내보았다.

오일러 회로에 의한 행렬표현으로 얻어질 수 있는 효과는 컴퓨터의 기억 용량의 절약이다. 기억 용량을 줄인다는 것은 모바일 시스템(mobile system)에서 의미를 가진다.

참고 문헌

1. 김성숙, “e의 역사적 기원과 의의,” 한국수학사학회지 제17권 제3호(2004), 33-42.
2. 신현성 · 임미선 · 이희종, 이산수학, 천재교육, 2004.
3. 이중권, “수학 탐구학습에서 지식 형성에 대한 연구,” 한국수학사학회지 제17권 제3호(2004), 109-120.
4. 유원희, 이산수학, 경문사, 1993.
5. 이주복 · 임호순, 이산수학, 정익사, 2003.
6. 이준열, “이산수학 제7차 교육과정의 구현 방안 연구,” 한국수학교육학회지 제41권

제1호(2002), 127-137.
7. 이재학, "중등 교사 양성을 위한 이산수학 강좌에 대한 연구," 한국수학교육학회지, 제42권 제4호(2003), 579-588.
8. 정치봉, 이산수학, 경문사, 1999.
9. 한길준, "학교에서의 이산수학과 그 역할에 관한 연구," 한국수학교육학회지, 제16권 제1호(2003), 45-62.
10. www.utc.edu.

A Historical Background of Graph Theory and the Computer Representation

Dept. of Computer Science, Anyang University **Hwa-jun Kim**
Dept. of Computer Science, Anyang University **Su-young Han**

This paper is aimed at studying a historical background of graph theory and we deal with the computer representation of graph through a simple example. Graph is represented by adjacency matrix, edge table, adjacency lists and we study the matrix representation by Euler circuit. The effect of the matrix representation by Euler circuit economize the storage capacity of computer. The economy of a storage capacity has meaning on a mobile system.

Key words: graph theory, incidence matrix, Euler circuit

2000 Mathematics Subject Classification: 97U20

ZDM Subject Classification: N79

논문 접수: 2004년 11월 9일, 심사 완료: 2004년 12월