10. 그래프 2

- 개요
  - 학습목표
    - 평면 그래프의 특징을 이해하고 주어진 그래프가 평면 그래프인지 판별할 수 있다.
    - 주어진 그래프에서 오일러 트레일과 오일러 투어를 찾아낼 수 있다.
    - 주어진 그래프에서 해밀턴 경로와 해밀턴 사이클을 찾아낼 수 있다.
    - 가중 그래프에서 최단경로를 찾는 알고리즘을 이해하고 활용할 수 있다.
  - 주요용어
    - 평면 그래프
    - 오일러의 공식
    - 4색정리
    - 오일러 트레일
    - 오일러 투어
    - 오일러 그래프
    - 해밀턴 경로
    - 해밀턴 사이클
    - 가중 그래프
    - 최단경로 문제
    - Dijkstra 알고리즘
    - 교착상태
- 1. 그래프의 탐색[245]
  - 평면 그래프(planar graph)
    - 평면 그래프
      - 그래프의 모든 변을 서로 교차하지 않게 그릴 수 있다면 평면 그래프이다.
    - 예제 10-1
      - 변을 조정하더라도 항상 하나 이상의 변이 서로 겹치게 된다.
    - 예제 10-2
      - 변을 조정하면 평면 그래프임이 확실해진다.
    - 면(face)
      - 변들의 사이클을 경계로 형성된 공간
      - 위 평면 그래프는 전체 공간을 네 개의 영역 f1, f2, f3, f4로 나누고 있음.
      - 네 개의 영역 중 f1, f2, f3는 그래프의 내부에 위치하고 있고 f4는 외부에 위치하고 잇다.
      - f4는 (1, 2, 4, 1)을 경계로 만들어진 면이다.
    - 오일러 공식 (Euler's Fomula)
      - 연결된 평면 그래프에서 꼭지점의 수를 v, 변의 수를 e, 면의 수를 f라고 할 때, $v - e + f = 2$이다.
      - 변의 수 e에 대해 수학적 귀납법을 통해서 증명할 수 있다.
        - 변의 개수를 하나 증가시키기 위해서 새로운 꼭지점을 추가한 뒤 연결하거나, 기존의 두 꼭지점을 연결시키는 방법으로 이루어진다.
      - 증명
        - 기본단계
          - $e = 0$ 이면 $v = 1$, $f = 1$ 이며 $v - e + f = 2$이다.
        - 귀납가정
          - $e = n$인 평면 그래프에 대해서 $v - e + f = 2$라고 가정한다.
        - 귀납단계
          - $e = n + 1$일 때 해당 수식이 만족하는 것을 보여야 한다.
          - 여기서 변의 개수를 1개 늘리는 방법은 다음과 같이 두 가지가 존재한다.
            - 새로운 꼭지점을 추가하여 그래프 내의 꼭지점과 연결하는 경우
              - 기존 변들과 교차하지 않으면서 새로운 꼭지점과 그래프 내의 꼭지점을 연결하게 되면, v, e는 1씩 증가하지만, f는 증가하지 않으므로 다음이 성립된다.
                - $(v+1) - (e+1) + f = v - e + f + 1 - 1 = 2$
            - 그래프 내의 두 꼭지점을 연결하는 경우

기존 변들과 교차하지 않으면서 그래프 내의 두 꼭지점을 연결하게 되면, v는 증가하지 않고 e, f는 1씩 증가하므로 다음이 성립된다.

$$v - (e + 1) + (f + 1) = v - e + f - 1 + 1 = 2$$

결국, e = n + 1인 평면 그래프에 대해서 v - e + f = 2이다. 따라서 모든 평면 그래프에 대해서 성립한다.

예제 10-3

4색 정리 (the four color theorem)

- **4색정리**
  - 지도의 인접한 구획을 서로 다른 색으로 칠하는 데 오직 4가지 색이면 충분하다는 정리
- 모든 지도는 평면 그래프로 표현할 수 있음
  - 지도의 각 구획은 꼭지점으로 대치한다.
  - 두 개의 구획이 맞닿아 있다면 꼭지점 사이에 변을 둔다.
- **4색 정리 (the four color theorem)**
  - 평면 그래프가 주어졌을 때, 각 꼭지점에 대하여 인접한 꼭지점과 다른 색으로 칠하는 데 필요한 색은 네 가지이면 충분하다.
  - 오일러 투어를 구하는 문제로 바꾸어 생각할 수 있음

1. 그래프의 탐색 (2) [249]

- 오일러 트레일(Eulerian trail)과 오일러 투어(Eulerian tours)
  - 오일러 트레일
    - 그래프의 모든 변들을 각각 한 번만 지나는 트레일
    - 오일러 트레일은 모든 변을 '정확히' 한 번만 지나가야 한다.
    - = 한붓그리기 = 오일러 사이클 = 오일러 회로(Eulerian circuit)
    - 오일러 그래프(Eulerian graph)
      - 오일러 투어를 갖는 그래프를 오일러 그래프(Eulerian graph)라고 한다.
  - 오일러 투어
    - 시작점과 종점이 동일한 오일러 트레일
    - **예제 10-4**
- 오일러 그래프(Euler graph) 정리
  - 오일러 그래프 정리
    - 연결 그래프가 오일러 투어를 가지기 위한 필요충분조건은 그래프의 모든 꼭지점의 차수가 짝수인 것이다.
      - 연결 그래프이고, 모든 꼭지점의 차수가 짝수라면 오일러 투어를 가진다.
      - (역도 성립) 그래프가 오일러 투어를 가진다면 그래프는 연결된 그래프이고 모든 꼭지점의 차수가 짝수이다.
    - 증명
      - 연결 그래프 G = (V, E)가 오일러 투어를 가진다고 가정하자. V는 오일러 투어에 참여하는 꼭지점들의 집합이므로, V의 임의의 꼭지점 v는 다른 꼭지점과 변을 통해 연결되어 있을 것이다.
      - v와 연결된 변들은 오일러 투어를 통해서 들어오는 변과 나가는 변으로 나누어 볼 수 있는데,
        - 이 때 들어오는 변의 수와 나가는 변의 수는 항상 동일하다.
        - 따라서 모든 꼭지점의 차수는 짝수이다.
    - 알고리즘
      - 단계 1: G의 임의의 꼭지점 v를 고른다.
      - 단계 2: v에서 시작하고 v에서 끝나는 임의의 사이클 C를 선택한다. (※ 인접한 임의의 꼭짓점 u에 착수 가능한 임의의 변으로 진입 때 하나의 변을 사용하더라도 또 하나의 변을 통해서 진출할 수 있다. 이 과정을 반복해서 수행하면 v를 향해 돌아올 수 있다.)
      - 단계 3: C가 오일러 투어이면 증명을 끝낸다. 만약 아니라면 아래 과정을 반복한다.
        - 단계 3-1: G에서 C에 속하는 모든 변을 제거하고 새로운 G' 를 만든다. (G의 모든 꼭짓점은 짝수의 차수를 가지고 있고, C의 모든 꼭짓점도 짝수의 차수를 가지고 있다. 따라서 새로 생성된 G'의 모든 꼭짓점도 짝수의 차수를 가진다.)
        - 단계 3-2: C와 G' 이 공유하는 꼭짓점 중 하나를 고르고 w라 한다. (G는 연결 그래프이므로 C와 G' 에는 적어도 하나의 꼭짓점을 공유한다.)
        - 단계 3-3: G'에서 w에서 시작하고 w에서 끝나는 임의의 사이클 C'을 선택한다.
        - 단계 3-4: 기존의 C와 새로 선택된 C'을 합쳐서 새로운 C'' 를 만든다. (w에서 C와 C'으로 가는 투어가 서로 만나므로, w에서 둘을 결합하여 하나의 투어로 만들 수 있다.)
      - 적용 [251 - 253]
- 해밀턴 경로(Hamilton path)와 해밀턴 사이클(Hamilton cycle)

    - 해밀턴 경로
      - 그래프의 모든 꼭지점을 각각 한 번씩만 지나는 경로를 해밀턴 경로라 한다.
    - 해밀턴 사이클
      - 시작점과 종점이 같은 해밀턴 경로를 해밀턴 사이클이라고 한다.
    - 오일러 투어가 그래프의 변에 대한 탐색방법이었다면, 해밀턴 사이클은 그래프의 꼭지점에 대한 탐색 방법이다.
    - 특징
      - 해밀턴 사이클이 존재한다는 것은 해밀턴 경로(꼭짓점이 다른것)가 존재한다는 것이지만, 역은 성립하지 않는다.
    - 예제 10-5
    - 예제 10-6

- 2. 그래프의 활용 [255]
  - 가중 그래프(weighted graph)와 가중치(weight)
    - 가중 그래프(weighted graph)
      - 그래프의 각 변에 실수가 붙여진 그래프
    - 가중치(weight)
      - 변에 부여된 값
    - 그래프 G의 서브그래프(subgraph) H에 대해 어떤 H의 모든 가중치의 합이 최소값을 가지는가에 대한 연구
      - 최단경로 문제
        - 출발지와 도착지를 정해 주었을 때 어떠한 경로가 가장 짧은 거리인지 찾는 문제
      - 최소 신장 트리 문제
        - 제 11장 트리에서 다룰 것
  - (1) 최단경로 문제
    - 다익스트라(Edsger W. Dijkstra)
      - 어떠한 변도 음수값의 가중치를 갖지 않는 방향 그래프에서 주어진 출발점과 도착점 사이의 최단경로를 찾아주는 알고리즘을 고안
      - function Dijkstra(G, w, s) { foreach vertex v in V[G] { d[v] ← ∞; } d[s] ← 0; Q ← V[G]; while Q ≠ ∅ do { d[u]의 값이 최소인 꼭짓점 u ∊ Q를 선택; Q ← Q - {u}; Q′ ← {v ∊ Q | v는 u와 인접} foreach vertex v in Q′ { d[v] ← min{ d[v], d[u] + w(u,v) }; } } }
        - G: 그래프 w(u,v): 간선 (u,v)의 가중치 s: 시작 정점 d[v]: 정점 v까지의 최단 거리 추정값 Q: 아직 최단 거리가 확정되지 않은 정점 집합
        - 이 알고리즘은 연결된 가중 그래프 G = (V, E)의 꼭지점 s에서부터 모든 꼭지점까지의 최단거리를 구한다.
        - w(i, j)
          - 꼭지점 i에서부터 꼭지점 j까지의 가중치
        - d[x]
          - 꼭지점 s에서부터 x까지의 거리
        - 수행이 모두 종료된 후에는
          - d[x]에 u부터 x까지의 최단경로의 길이가 들어가게 된다.
      - 특징
        - 음의 가중치가 없는 그래프에서 한 정점에서 모든 정점까지의 최단 경로를 구할 때 사용
        - 이 구현은 우선순위 큐 없이 선형 탐색 기반으로 되어 있어 시간복잡도는 O(V^2)
        - 우선순위 큐(힙 구조 등)를 사용하면 O((V+E)log V)로 개선됨
    - 다익스트라 적용 [256 - 261]
      - 강의록
      - 문제

- 2. 그래프의 활용 (2) [261]
  - (2) 교착상태 찾기
    - 교착상태(deadlock)
      - 둘 이상의 프로세스가 서로 상대방의 프로세스 결과를 무한정 기다리는 상황
      - 사례
        - 운영체제의 교착상태를 해소하기 위해, 테이프 드라이브를 사용자 1에서 제거하거나 프린터를 사용자 3에서 제거하여야 한다.
    - 교착상태를 발견하는 방법은 방향 그래프에서 방향 사이클을 발견하는 알고리즘과 동일하다.
      - 절차

- 1. 진출 차수가 0인 꼭지점을 찾는다.
- 2. 진출 차수가 0인 꼭지점에 들어오는 변을 전부 제거한다.
- 3. 진입 차수가 0인 꼭지점을 찾는다.
- 4. 진입 차수가 0인 꼭지점에서 밖으로 나가는 변을 전부 제거한다.
- 5. 남아 있는 꼭지점이 방향 사이클에 포함되는지 판단한다.

- 원리
  - (1) 진입하는 변이 있지만 진출하는 변이 없는 꼭지점이나 (2) 진출하는 변이 있지만 진입하는 변이 없는 꼭지점은 사이클에 포함될 수 없기에 고려 대상에서 제외한다.
  - 이런 과정을 거치고 나서 그래프 내에 방향 사이클을 발견했다면 교착상태를 찾아낸 것이다.
- 위 과정을 거쳐 [그림 10-17]에서 발견한 방향 사이클은 다음과 같음.
- 방향 그래프의 모든 꼭지점이 방향 사이클에 포함되지 않은 경우, 방향 그래프에는 교착상태가 존재하지 않는다.
  - 위 그래프는 교착상태가 발생하지 않음

연습문제
- 5
- 6
- 7
- 8
- 13
- 14