

알고리즘 과제 2

컴퓨터과학과 2016010873 박정욱

프로그램 개요

프로그램 내용 및 범위

프로그램 목표

본 프로그램은 가중치가 주어진 간선과 정점으로 만들 수 있는 최소 신장 트리(Minimum Spanning Tree)를 구하여, 그 결과를 도식화한 것을 보여주는 것이 주 목표이다.

프로그램 내용

- 입력 기능
 - 그래프의 정점을 입력한다.
 - 그래프의 간선과 그 가중치를 입력한다.
- 최소 신장 트리 계산
 - Prim 알고리즘을 이용하여 최소 신장 트리를 구한다.
 - Kruskal 알고리즘을 이용하여 최소 신장 트리를 구한다.
- 결과 출력
 - 주어진 정점 개수의 절반만큼 알고리즘을 진행한 시점의 형태를 출력한다.
 - 주어진 정점이 모두 연결되어, 최소 신장 트리가 완성되는 시점의 형태를 출력한다.

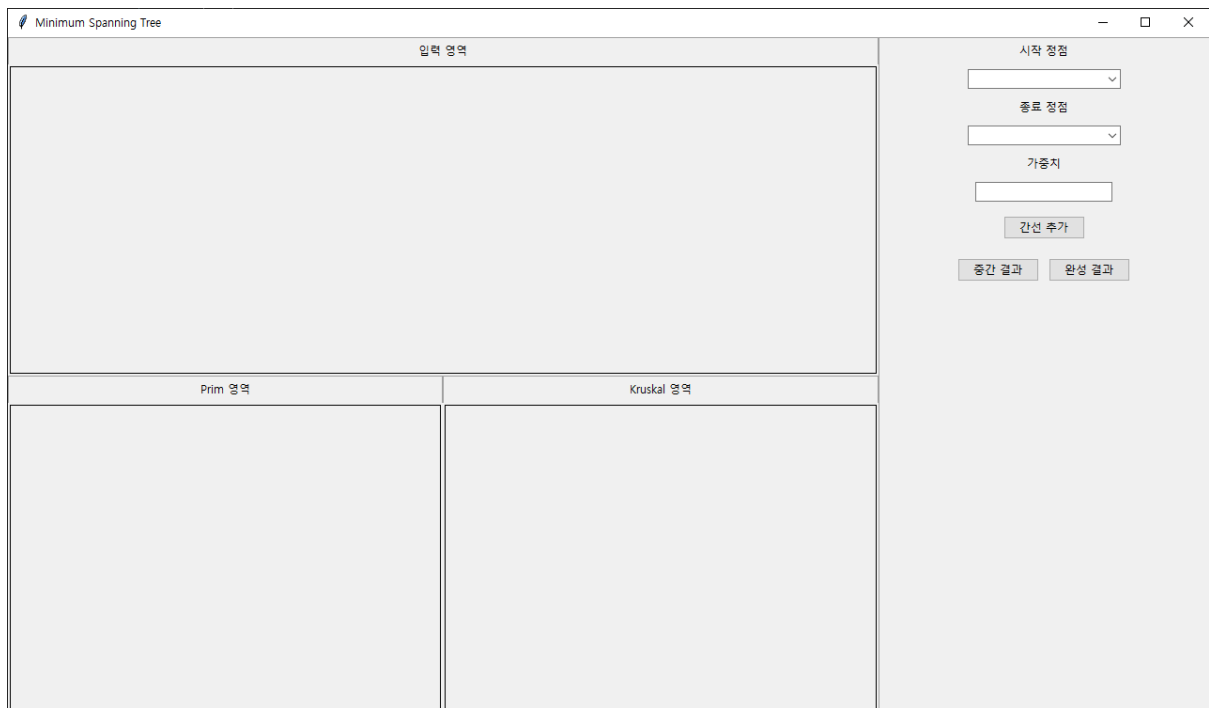
프로젝트 구현 환경 및 설계 제한 요소

- 구현 환경
 - 플랫폼 : Windows 10
 - 언어 : Python
 - 개발 도구 : Visual Studio Code
 - 형상 관리 도구 : Git, GitHub

- 설계 제한 요소

- 임의의 간선을 표시한 그림이 임의의 정점을 표시한 그림과 겹쳐지는 경우, 해당 간선은 유효하지 않은 것으로 판단한다.
- 두 개의 정점 사이에 이미 간선이 존재할 경우, 해당 간선은 유효하지 않은 것으로 판단한다.

사용자 관점에서의 설계



다음의 요소를 포함한다:

- 정점을 입력할 수 있는 영역
- 시작 정점과 종료 정점, 가중치를 입력하여 간선을 추가할 수 있는 영역
- 최소 신장 트리를 중간 단계 또는 최종 단계까지 계산하는 명령을 내릴 수 있는 영역
- Prim 알고리즘을 이용해 계산된 최소 신장 트리를 도식화하여 보여주는 영역
- Kruskal 알고리즘을 이용해 계산된 최소 신장 트리를 도식화하여 보여주는 영역

기능적 관점에서의 설계

정점 입력

1. 정점을 입력받을 수 있는 Canvas 영역에서 마우스 클릭 이벤트 발생 시 얻을 수 있는 클릭 위치의 x, y 좌표를 임시로 저장한다. 이후, 내부 카운터 변수를 이용해 정점의 식별자를 부여한다. 마지막으로, 임시 저장한 좌표와 부여해준 식별자를 속성으로 가지는 Vertex 객체를 생성하여 내부 리스트에 추가한다.
 2. 시작 정점과 종료 정점을 선택하는 ComboBox를 갱신한다. 이때, 각 Combobox에 선택된 값이 존재할 경우, 반대쪽 ComboBox 목록에 있는 해당 값을 제거한다. 예를 들자면 시작 정점 ComboBox에 '1'이라는 값이 선택되어있을 때, 종료 정점 ComboBox 목록에 있는 '1'을 제거하는 형식이다.
 3. 저장된 정점을 이용해 정점을 Canvas 영역에 표시한다. 현재 사용중인 라이브러리 tkinter(이하 tkinter)는 타원을 그리는 함수만을 지원하기 때문에, 정점의 좌표와 임의의 반지름을 입력받아 원을 계산하는 별도의 함수를 정의하여 원을 그리고, 그 위에 식별자 텍스트를 그린다.
- 입력: 정점 입력 Canvas 영역 내 클릭 위치의 x 좌표 및 y 좌표

간선 입력

1. 시작 정점과 종료 정점을 선택하는 Combobox에 값을 지정해준다. 이때, 선택된 값을 반대쪽 ComboBox 목록에서 제거한다. **정점 입력**의 2번과 같은 형식으로 이해할 수 있다.
2. '간선 추가' 버튼을 클릭하면 입력받은 시작/종료 정점을 이용해 선분을 그리고, 선분의 중점을 계산하여 해당 좌표에 가중치 텍스트를 그린다. 이때, 정점 또는 가중치가 입력되지 않았다면 에러 메시지박스를 보여준다.
3. 간선과 정점의 충돌 여부를 판정한다.
 - tkinter는 Canvas 영역 내에 그려진 객체들의 Bounding Box(경계 상자)를 반환하는 함수를 제공한다. 이를 이용하여 해당 간선과 충분히 가까이 있는 정점들을 일차적으로 검출한다.
 - 검출된 정점들에 대하여, 원과 직선의 위치 관계를 이용하여 추가적인 충돌 여부를 판정한다. 이때 판별식 D 를 계산하는 함수를 별도로 정의하여, D 의 값이 0 이상일 경우 정점과 간선이 충돌하는 것으로 판단한다.

- 시작 정점과 종료 정점을 제외한 모든 정점에 대하여, 하나의 정점이라도 간선과 충돌한다면 Canvas 영역에 그려진 선분과 가중치 텍스트를 제거하고 에러 메시지박스를 보여준다.
4. 간선의 중복 여부를 판정한다. 이전까지 저장되어있는 모든 간선의 시작 정점과 종료 정점에 대하여, Canvas 영역에 그려진 선분의 시작점과 끝점과 서로 같다면 선분과 가중치 텍스트를 제거하고 에러 메시지박스를 보여준다.
 5. 입력된 간선이 충돌 여부 판정과 중복 여부 판정을 통과하였다면 해당 간선을 내부 리스트에 저장한다.
- 입력: 시작 정점, 종료 정점 및 가중치

최소 신장 트리 알고리즘 계산

1. '중간 결과' 버튼 클릭 시, 정점 개수의 절반만큼 알고리즘을 반복하여 진행하고 알맞은 영역에 계산된 그래프를 도식화하여 그린다.
 2. '완성 결과' 버튼 클릭 시, 알고리즘을 최종 단계까지 진행하고 알맞은 영역에 계산된 그래프를 도식화하여 그린다.
- 위 그래프를 그릴 때, 입력 영역 Canvas와 결과 영역 Canvas의 가로 길이가 다르다는 문제가 있다. 따라서 각각의 가로 길이에 정점과 간선의 x좌표 위치가 비례하도록 계산하여 그린다.

정보적 관점에서의 설계

Vertex: 정점을 표현하는 클래스

- 속성
 - x좌표 - int
 - y좌표 - int
 - 식별자 - int

Edge: 간선을 표현하는 클래스

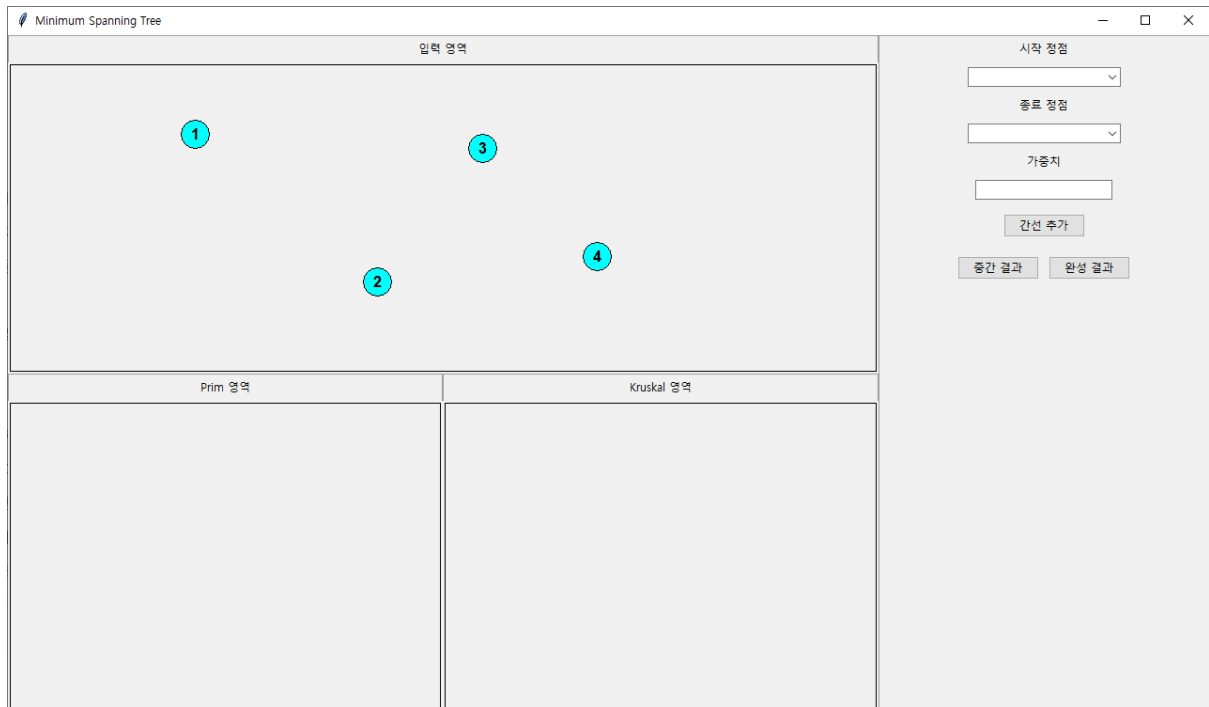
- 속성
 - 시작 정점 - Vertex
 - 종료 정점 - Vertex
 - 가중치 - int

Mst: 최소 신장 트리를 계산하기 위한 정보를 담고 있는 클래스

- 속성
 - 정점 식별자 부여를 위한 내부 카운터 - int
 - 정점 리스트 - list
 - 간선 리스트 - list

실행 화면

정점 그리기



Minimum Spanning Tree

입력 영역

Prim 영역

Kruskal 영역

시작 정점

4

종료 정점

3

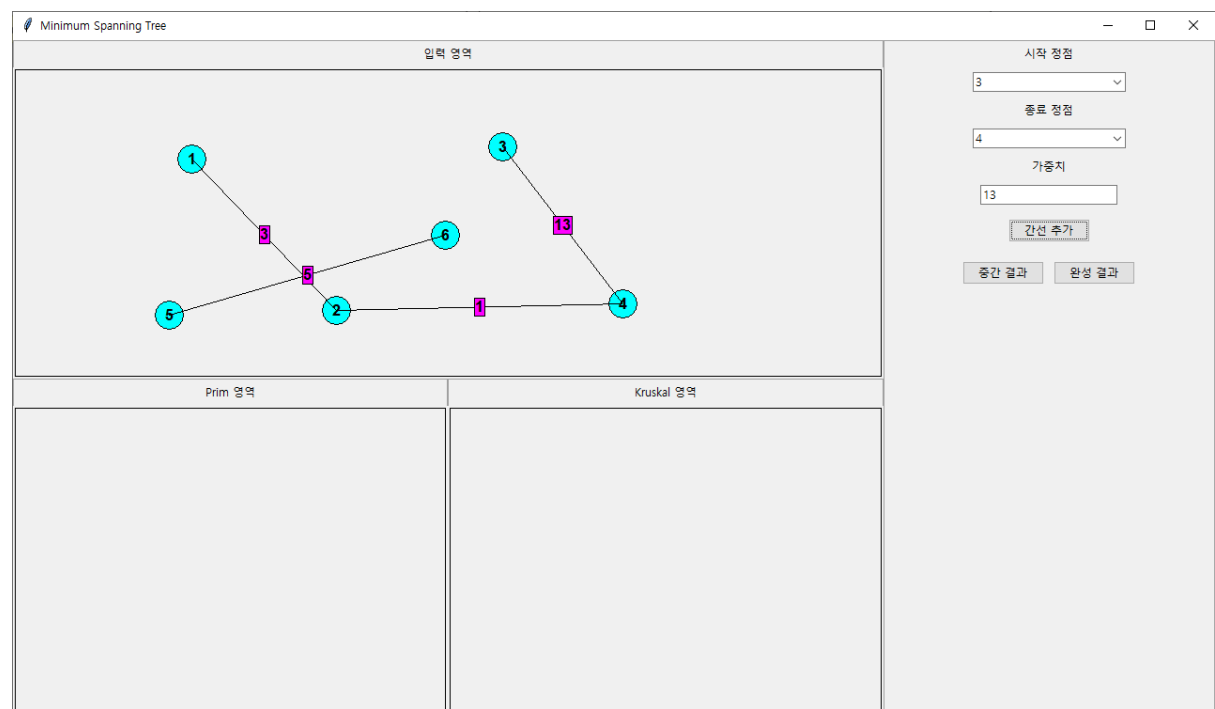
가중치

5

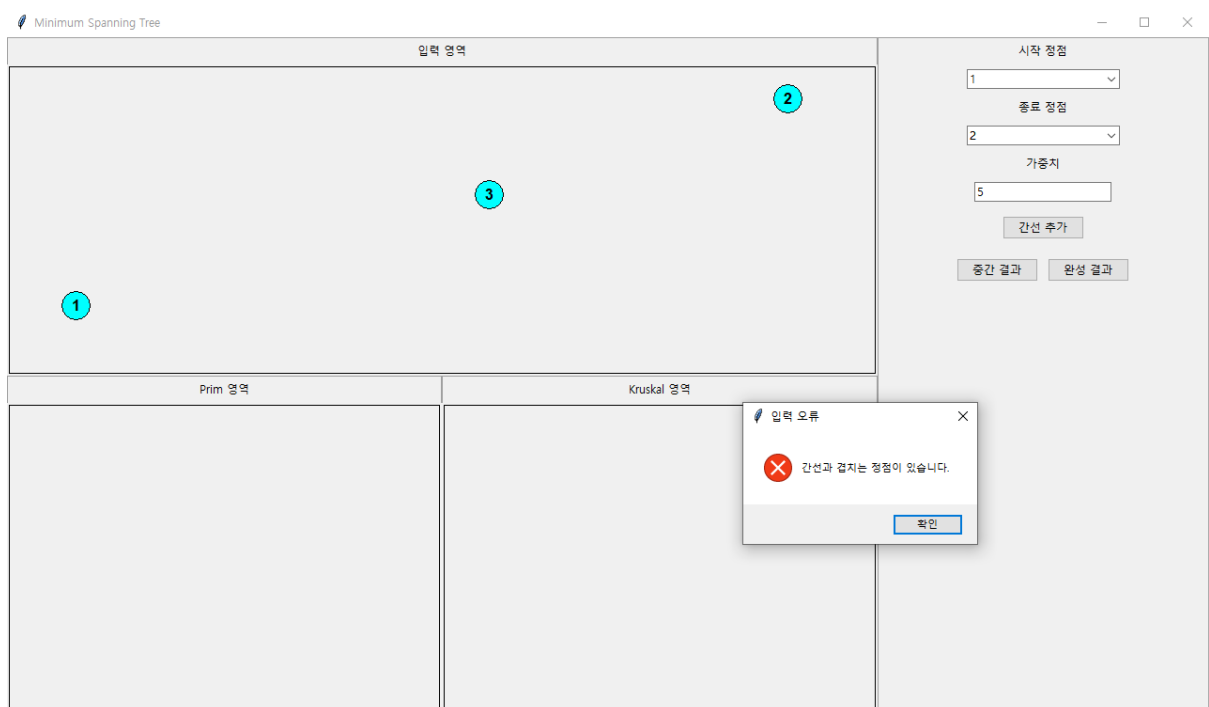
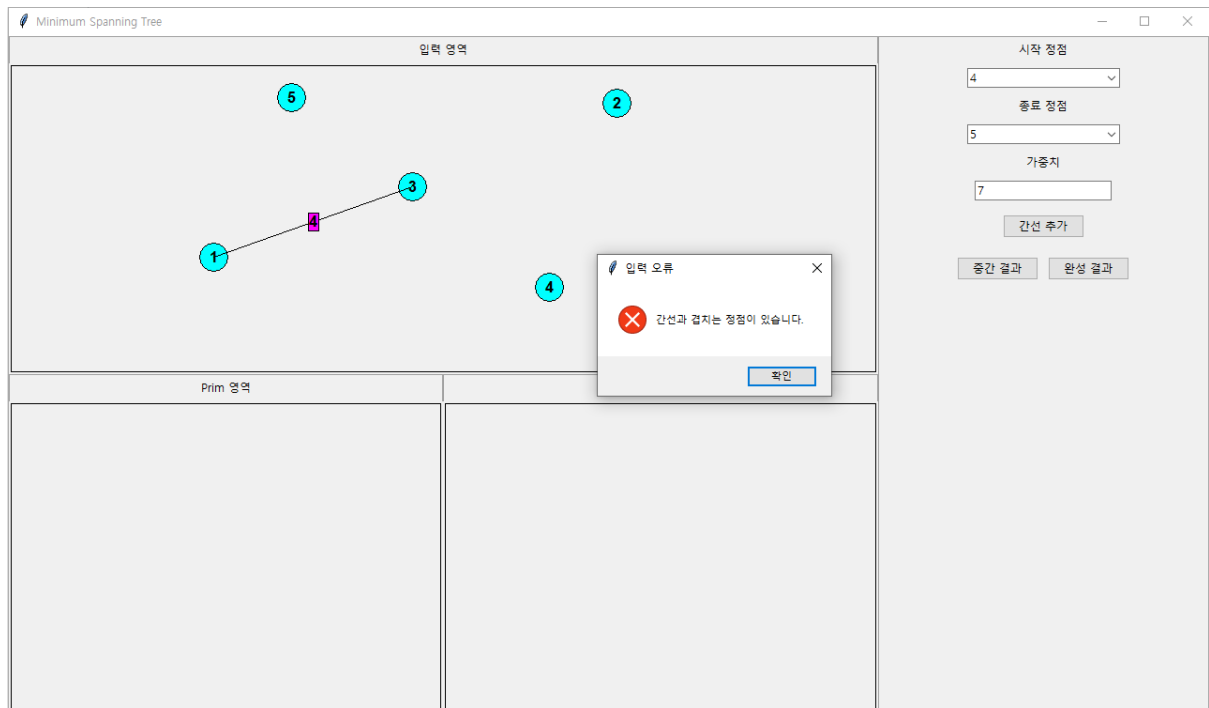
간선 추가

중간 결과

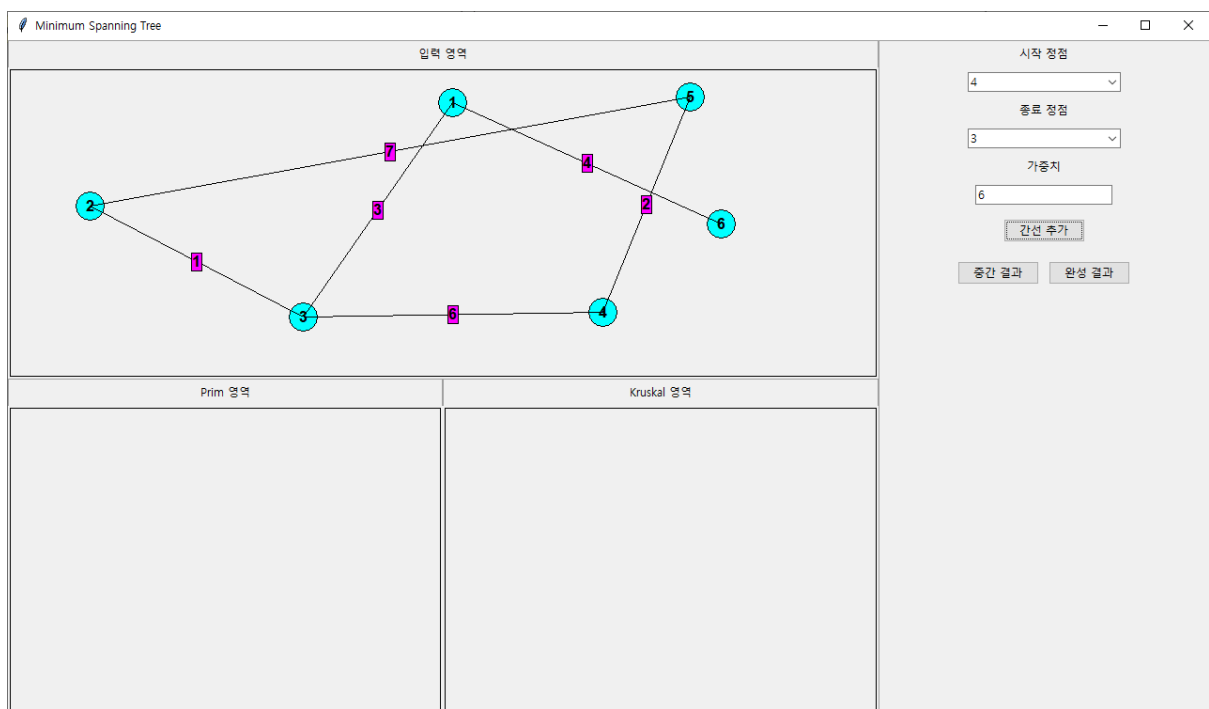
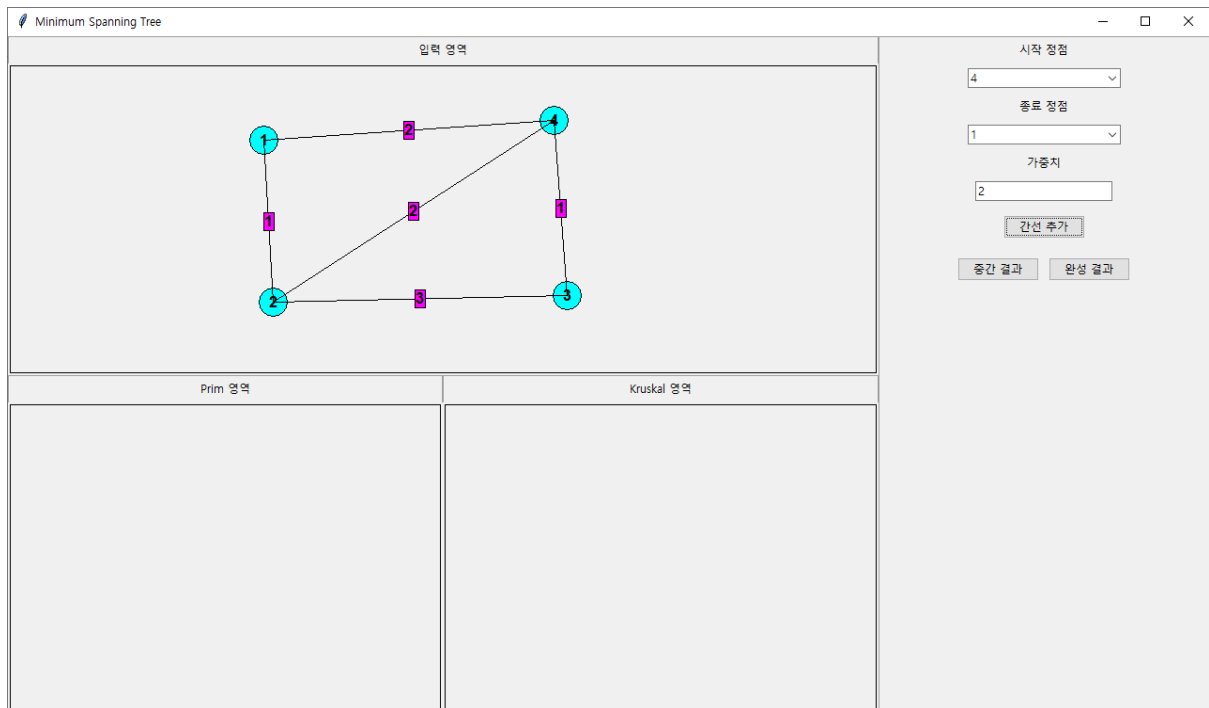
완성 결과



간선 그리기 불가 시



완성 화면



최소 신장 트리 실행 화면

