

Data Structure Assignment #3

Kruskal's algorithm



과목명	자료구조
담당 교수	석흥일 교수님
제출일	2019년 05월 31일
학과	컴퓨터학과
학번	2018320161
이름	송대선

1. 개발환경:

window 10 os에서 Visual Studio 2019로 C언어를 이용하여 개발하였습니다.

2. 기능설명:

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.):
```

처음에 프로그램을 실행시키면 생성할 노드의 개수를 입력하는 창이 나옵니다.

여기서 자신이 원하는 노드의 개수를 정수값으로 입력합니다.

숫자를 제대로 입력하지 않으면 오류가 날 수도 있습니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 0  
Node는 1개 이상이어야 합니다!
```

```
C:\Users\Wpc\source\repos\DataStructureAssignment_3\Debug\DataStru  
종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요.
```

1개 이상의 값을 입력하지 않으면 오류라고 표시해주고 프로그램을 종료합니다.

그래프는 기본적으로 1개 이상의 노드를 가져야 합니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5  
Node 0, 1, 2, 3, 4이 만들어 졌습니다.  
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.):
```

노드 생성 이후 간선을 생성합니다.

여기서 자신이 원하는 간선의 개수를 정수값으로 입력합니다.

숫자를 제대로 입력하지 않으면 오류가 날 수도 있습니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5  
Node 0, 1, 2, 3, 4이 만들어 졌습니다.  
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): -1  
Edge는 0개 이상이어야 합니다!
```

```
C:\Users\Wpc\source\repos\DataStructureAssignment_3\Debug\DataStru  
해 종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요.
```

0개 이상의 값을 입력하지 않으면 오류라고 표시해주고 프로그램을 종료합니다.

그래프는 기본적으로 음수 개의 간선을 가질 수 없습니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5  
Node 0, 1, 2, 3, 4이 만들어 졌습니다.  
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6  
1번째 edge (node1/node2/weight):
```

원하는 간선의 개수를 입력한 뒤에 간선의 정보를 입력합니다.

예를 들어 node 0과 node 2 사이에 weight가 20인 간선을 만들고 싶다면,

“0/2/20” 또는 “2/0/20”으로 입력해 줍니다.

형식대로 제대로 입력하지 않으면 오류가 날 수도 있습니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5
Node 0, 1, 2, 3, 4이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6
1번째 edge (node1/node2/weight): 5/6/3
존재하는 Node들만을 입력하여 주십시오
C:\Users\wpc\source\repos\DataStructureAssignment_3\Debug\DataStr
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

edge는 “Node 0, 1, 2, 3, 4이 만들어 졌습니다.”

에서 말해주는 생성된 노드 0, 1, 2, 3, 4을 이용해서만 만들어야 합니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5
Node 0, 1, 2, 3, 4이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6
1번째 edge (node1/node2/weight): 0/0/1
self edge는 고려하지 않습니다.
C:\Users\wpc\source\repos\DataStructureAssignment_3\Debug\DataStr
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

간선은 self edge가 되면 안 됩니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5
Node 0, 1, 2, 3, 4이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6
1번째 edge (node1/node2/weight): 0/2/4
2번째 edge (node1/node2/weight): 2/0/6
중복되는 edge는 허용하지 않습니다.
C:\Users\wpc\source\repos\DataStructureAssignment_3\Debug\DataStr
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

multigraph는 허용하지 않습니다.

```
Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5
Node 0, 1, 2, 3, 4이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6
1번째 edge (node1/node2/weight): 1/0/-3
weight는 음수가 아닌 수여야 합니다.
C:\Users\wpc\source\repos\DataStructureAssignment_3\Debug\DataSt
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

edge의 weight는 non-negative한 값이어야 합니다.

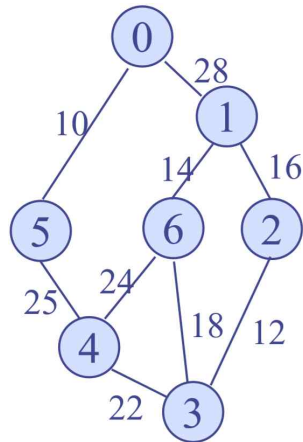
```

Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 5
Node 0, 1, 2, 3, 4이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 6
1번째 edge (node1/node2/weight): 0/1/12
2번째 edge (node1/node2/weight): 0/3/8
3번째 edge (node1/node2/weight): 4/0/2
4번째 edge (node1/node2/weight): 1/3/5
5번째 edge (node1/node2/weight): 4/1/1
6번째 edge (node1/node2/weight): 3/4/7
Graph가 Connected되지 않았습니다.
C:\Users\wpc\source\repos\DataStructureAssignment_3\Debug\DataSt
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

spanned tree를 찾는 것은 주어진 그래프가 이미 connected 되었다는 가정이 있으므로, 사용자가 입력한 그래프가 connected graph인지를 확인해 줍니다.

Kruskal's algorithm을 위한 예시 input를 다음과 같이 정의합니다.



node의 개수: 7
edge의 개수: 9
0/5/10
0/1/28
1/6/14
1/2/16
5/4/25
6/4/24
6/3/18
2/3/12
4/3/22

```

Node를 몇개 만드시겠습니까? (Node는 1개 이상이어야 합니다.): 7
Node 0, 1, 2, 3, 4, 5, 6이 만들어 졌습니다.
Edge를 몇개 만드시겠습니까? (Edge는 0개 이상이어야 합니다.): 9
1번째 edge (node1/node2/weight): 0/5/10
2번째 edge (node1/node2/weight): 0/1/28
3번째 edge (node1/node2/weight): 1/6/14
4번째 edge (node1/node2/weight): 1/2/16
5번째 edge (node1/node2/weight): 5/4/25
6번째 edge (node1/node2/weight): 6/4/24
7번째 edge (node1/node2/weight): 6/3/18
8번째 edge (node1/node2/weight): 2/3/12
9번째 edge (node1/node2/weight): 4/3/22

Spanned Tree의 edges는 다음과 같습니다.
(Node1, Node2)->weight
(0, 5)->10
(2, 3)->12
(1, 6)->14
(1, 2)->16
(3, 4)->22
(4, 5)->25

C:\Users\pc\source\repos\DataStructureAssignment_3\Debug\DataSt
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

프로그램이 Kruskal's algorithm을 이용하여 spanned tree에 필요한 edge들을 출력합니다.
 예를 들어 "(2, 3)->12"는 node 2, 3사이에 weight가 12인 edge를 의미합니다.

