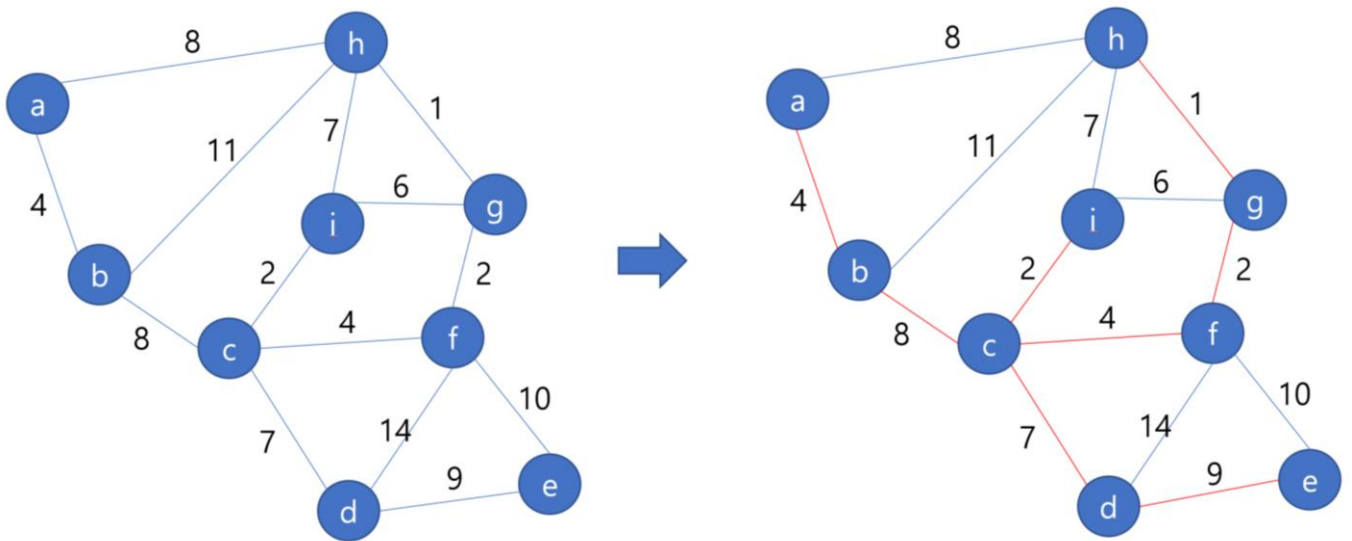


## Assignment #2

### Kruskal Algorithm using Heap and Disjoint Set

#### 과제 설명

Kruskal 알고리즘을 Heap과 Disjoint Set을 사용하여 구현하시오.



- 주어진 Graph를 기준으로 Minimum Spanning Tree (MST)를 구한다.
- Disjoint set은 이론시간에 다른 코드를 그대로 사용하도록 한다.
- Heap은 자유롭게 구현하도록 한다. Heap을 통해서 배열에 들어간 값을 sort 한다.
- 같은 weight를 가지는 edges가 있는 경우 여러 개의 MST가 나올 수 있다. 대신 total weight의 값은 같아야 한다.

#### 필수 제출 내용

- "heap.c", "graph.c" 파일
- 보고서 (ReportTemplate.docx 파일 완성)

## 과제 목표

### 1. "heap.c" 완성

해당 파일의 `heap_sort()` 함수를 구현한다. Heap의 개념을 사용하여 파라미터로 주어진 간선 배열을 오름차순으로 정렬해야 한다.

- 각 간선의 비교에는 "heap.h" 파일에 정의된 `compare()` 함수를 사용한다.
- Heap 정렬은 자유롭게 구현해도 무관하다.
  - 주어진 뼈대를 사용해 `build_heap`과 `heapify`를 사용해 구현하는 방법
  - 혹은 본인이 원하는 구현 방법
- Heap 정렬 이외에 다른 정렬 방법은 실행 결과와 무관하게 감점

### 2. "graph.c" 완성

해당 파일의 주어진 함수들을 완성하여 Disjoint Set 자료구조와 Kruskal MST 알고리즘을 구현한다.

- Disjoint Set은 강의노트에 첨부된 수도 코드를 구현하는 것을 권장한다.
- `find_minimum_spanning_tree()` 함수는 주어진 그래프의 MST를 그래프 구조체 형태로 저장하여 반환한다. MST를 구성하는  $(V-1)$ 개의 간선을 모두 포함해야 한다.
- 각 알고리즘을 정확히 구현하지 않으면, 시간제한 초과 혹은 메모리 제한 초과로 인해 일부 데이터들이 오답처리 될 수 있다.

## 과제 유의사항

- "heap.c"와 "graph.c" 파일만을 수정하여 과제를 수행한다.
- 모든 정점번호, 간선번호, 배열의 인덱스는 0부터 시작하는 것을 기준으로 한다.
- 주어진 헤더파일들을 수정하지 않는다. 채점에는 처음 주어진 초기 상태의 헤더파일을 사용하므로, 헤더 파일을 수정하여 문제를 해결한 경우 채점 시 불이익이 생길 수 있다.
- 디버깅 시에는 `printf`나 `puts`와 같은 콘솔 출력은 자유롭게 사용하여도 무관하다.
- 파일 입/출력 형식을 변경하는 경우 오답처리 된다.
- 실제 채점 시에는 샘플 데이터가 아닌 비공개 데이터로 채점을 진행한다.

## 입력 파일 설명

1. 파일 이름은 "input.txt"다.
2. 파일의 첫 줄에는 정점의 수  $V$  와 간선의 수  $E$  가 공백으로 구분되어 주어진다.
3. 모든 정점은  $0 \sim (V-1)$ 번의 인덱스를 가지며, 간선은  $0 \sim (E-1)$ 의 인덱스를 가진다.
4. 이후 한 줄에 하나씩 간선의 정보가 입력으로 주어진다. 간선의 정보는 두 노드의 번호와 가중치가 공백으로 구분되어 주어진다.
5. 간선의 수와 정점의 수는 최대 2 이상 20 만이하의 자연수로 주어진다.
6. 간선의 가중치는 0 이상 1,000 이하의 정수다.

## 출력 파일 설명

1. 파일 이름은 "output.txt"다.
2. 계산한 MST 를 입력 파일과 동일한 형식으로 출력한다.
3. 존재하지 않는 간선을 출력하거나 MST 가 되지 않는 간선을 출력하는 경우 오답으로 처리한다.
4. 이후 파일의 마지막에 가중치의 합을 출력한다.
5. **시스템상으로 자동 채점을 수행하므로 임의로 출력 형식을 변경하지 않는다.**

## 평가 기준

1. 힙 정렬을 정확히 구현할 수 있는가?
2. Disjoint set 자료구조를 이해하고 Union / Find 연산을 정확히 구현할 수 있는가?
3. MST 의 성질을 이해하고 Kruskal 알고리즘을 구현할 수 있는가?
4. 알고리즘을 정확하고 효율적으로 수행될 수 있도록 구현할 수 있는가?
5. 본인이 작성한 내용을 주석으로 충분히 설명할 수 있는가?