1. (**실습**: Kruskal's MST 알고리즘) 다음은 Kruskal의 MST 알고리즘을 구현한 코드이다. 아래 코드를 입력하고 실행하면서 Kruskal MST 알고리즘 구현법을 학습하시오. (Reference: https://en.wikipedia.org/wiki/Kruskal's_algorithm, CC-BY-SA, Reference: p.291 in (Horowitz et al., 1993), https://algs4.cs.princeton.edu/43mst/KruskalMST.java.html, GPLv3)

   - **실습** #1: 아래 코드에 발견된 최소신장트리의 총 비용을 출력하는 코드를 추가하시오.

```
public class Test {
	static class Edge implements Comparable<Edge>{
		int		v1,v2;
		double		weight;
		public Edge(int v1, int v2, double weight) {
			this.v1=v1;
			this.v2=v2;
			this.weight=weight;
		}
		@Override
		public String toString() {
			return "("+v1+","+v2+","+weight+")";
		}
		@Override
		public int compareTo(Edge that) {
			return this.weight<that.weight? -1 : this.weight>that.weight? 1 : 0;
		}
	}
	public static void main(String[] args) {
		String		input="0 1 70 0 3 10 1 2 80 3 2 50 3 4 40 2 5 20 4 2 30 4 5 60"; // for Kruskal's
		int		V=6;
		LinkedList<Edge> adjList[]=new LinkedList[V];
		for (int i = 0; i < adjList.length; i++) adjList[i]=new LinkedList<>();
		String		s[]=input.split("\\s+");
		for (int i = 0; i < s.length; i+=3){
			int		v1=Integer.parseInt(s[i]), v2=Integer.parseInt(s[i+1]);
			double		weight=Double.parseDouble(s[i+2]);
			adjList[v1].add(new Edge(v1,v2,weight));
			adjList[v2].add(new Edge(v2,v1,weight));
		}
		System.out.println(KruskalMST(adjList, V));
	}
	private static LinkedList<Edge> KruskalMST(LinkedList<Edge>[] adjList, int V) {
		LinkedList<Edge>mst=new LinkedList<>();
		LinkedList<Edge>edges=new LinkedList<>();
		for (int i = 0; i < adjList.length; i++) edges.addAll(adjList[i]);
		PriorityQueue<Edge>		minPQ=new PriorityQueue<>(edges);
		UF		uf=new UF(V);
		while(mst.size()<V-1 && minPQ.size()>0){
			Edge		e=minPQ.remove();
			if(uf.find(e.v1)==uf.find(e.v2)) continue;
			mst.add(e);
			uf.union(e.v1, e.v2);
		}
		return mst;
	}
}
public class UF {
	int		parent[], rank[], count;
	public UF(int N) {
		parent=new int[N];
		rank=new int[N];
		for (int i = 0; i < parent.length; i++){
			parent[i]=i;
		}
	}
	public void union(int i, int j) { // union by rank
		i=find(i);
		j=find(j);
		if(i==j) return;
		if(rank[i]<rank[j]) parent[i]=j;
		else if(rank[i]>rank[j]) parent[j]=i;
		else{
			parent[i]=j;
			rank[j]++;
		}
	}
	public int find(int i) { // path compression
		if(i!=parent[i]) parent[i]=find(parent[i]);
		return parent[i];
	}
	@Override
	public String toString() {
		return Arrays.toString(parent);
	}
}
```

2. (Prim's MST 알고리즘) 다음은 Prim의 MST 알고리즘을 구현한 코드이다. 아래 코드를 입력하고 실행하면서 Prim MST 알고리즘 구현법을 학습하시오. (Reference: https://en.wikipedia.org/wiki/Prim's_algorithm, CC-BY-SA, Reference: p.293 in (Horowitz et al., 1993), https://algs4.cs.princeton.edu/43mst/PrimMST.java.html, GPLv3)

- **실습** #1: 아래 코드는 현재까지 발견된 mst로부터 mst 외부의 동일 정점으로 향하는 모든 간선들을 우선순위큐에서 관리하도록 하고 있어 개선 필요.

```
public class Test {
	static class Edge implements Comparable<Edge>{
		int		v1,v2;
		double	weight;
		public Edge(int v1, int v2, double weight) {
			this.v1=v1;
			this.v2=v2;
			this.weight=weight;
		}
		@Override
		public String toString() {
			return "("+v1+","+v2+","+weight+")";
		}
		@Override
		public int compareTo(Edge that) {
			return this.weight<that.weight? -1 : this.weight>that.weight? 1 : 0;
		}
	}
	public static void main(String[] args) {
		String	input="0 1 40 0 3 10 1 2 70 3 2 80 3 4 20 2 5 30 4 2 50 4 5 60"; // for Prim's
		int		V=8;
		LinkedList<Edge> adjList[]=new LinkedList[V];
		for (int i = 0; i < adjList.length; i++) adjList[i]=new LinkedList<>();
		String	s[]=input.split("\\s+");
		for (int i = 0; i < s.length; i+=3){
			int		v1=Integer.parseInt(s[i]), v2=Integer.parseInt(s[i+1]);
			double	weight=Double.parseDouble(s[i+2]);
			adjList[v1].add(new Edge(v1,v2,weight));
			adjList[v2].add(new Edge(v2,v1,weight));
		}
		System.out.println(PrimMST(adjList, V));
	}
	private static LinkedList<Edge> PrimMST(LinkedList<Edge>[] adjList, int V) {
		LinkedList<Edge>	mst=new LinkedList<>();
		PriorityQueue<Edge>	minPQ=new PriorityQueue<>(adjList[0]);
		boolean	mstV[]=new boolean[V];
		mstV[0]=true;
		while(mst.size()<V-1 && minPQ.size()>0){
			Edge	e=minPQ.remove();
			if(mstV[e.v2]) continue;
			mst.add(e);
			mstV[e.v2]=true;
			for (Edge outEdge : adjList[e.v2]) if(mstV[outEdge.v2]==false) minPQ.add(outEdge);
		}
		return mst;
	}
}
```

References

- C로 쓴 자료구조론 (Fundamentals of Data Structures in C, Horowitz et al.). 이석호 역. 사이텍 미디어. 1993.
- 쉽게 배우는 알고리즘: 관계 중심의 사고법. 문병로. 한빛아카데미. 2013.
- C언어로 쉽게 풀어 쓴 자료구조. 천인국 외 2인. 생능출판사. 2017.
- 김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.
- 남궁성. 자바의 정석. 도우출판.
- 김윤명. (2010). 뇌를 자극하는 JSP & Servlet. 한빛미디어.