

cpp

```
void kruskal(BasicGraph& graph, BasicGraph& mst) {
    // 먼저, 그래프를 복사하고 간선을 제거합니다.
    // 이렇게 하면 나중에 MST 간선을 추가할 수 있습니다.
    mst = graph;
    mst.clearEdges();

    // 각 정점을 '클러스터'에 넣습니다. 초기에는 각 정점이 자신만 포함합니다.
    Map<Vertex*, Set<Vertex*>*> clusters;
    Set<Vertex*> allVertices = graph.getVertexSet();
    Vector<Set<Vertex*>*> allSets; // 나중에 메모리 해제를 위해
    for (Vertex* v : allVertices) {
        Set<Vertex*>* set = new Set<Vertex>();
        set->add(v);
        clusters[v] = set;
        allSets.add(set);
    }

    // 모든 간선을 가중치에 따라 우선순위 큐에 넣습니다.
    PriorityQueue<Edge> pq;
    Set<Edge*> allEdges = graph.getEdgeSet();
    for (Edge* edge : allEdges) {
        pq.enqueue(edge, edge->cost);
    }

    // 반복적으로 최소 가중치 간선을 큐에서 꺼내서 MST에 추가합니다.
    // 만약 간선의 끝점이 이미 연결되어 있지 않다면.
    Set<Edge*> mstEdges;
    while (!pq.isEmpty()) {
        Edge* e = pq.dequeue();
        Set<Vertex> set1 = clusters[e->start];
        Set<Vertex> set2 = clusters[e->finish];
        if (set1 != set2) {
            mstEdges.add(e);
            // 두 집합을 합칩니다.
            set1->addAll(*set2);
            for (Vertex* v : *set1) {
                Set<Vertex> setv = clusters[v];
                if (setv != set1) {
                    clusters[v] = set1;
                }
            }
        }
    }

    // MST 간선을 그래프에 추가합니다.
    // 직접 간선 포인터를 추가할 수는 없으므로
    // 나중에 메모리 해제하는 데 문제가 생길 수 있습니다.
    for (Edge* edge : mstEdges) {
        mst.addEdge(edge->start->name, edge->end->name, edge->cost, false);
    }
}
```

Table

Column 1	Column 2
Cell A	Cell B
Cell C	Cell D

Image with Bounding Box

 <!-- Normalization:
[0,0,1000,1000] -->