알고리즘 설계 및 분석 HW3 보고서

## 20192134 컴퓨터공학과 김은원

1. 실험 환경

OS: window 10 home

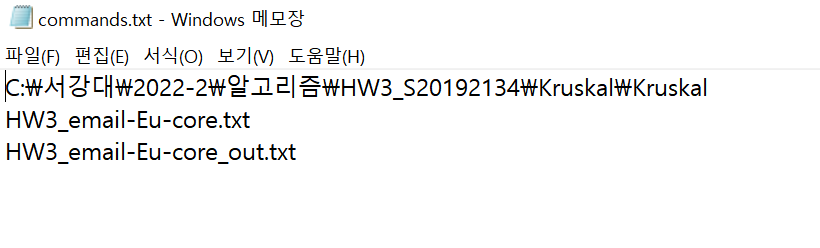
CPU: Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz

RAM: 8GB 2400Mhz DDR4

Compiler: Visual Studio 2022 Release Mode

1. 실험 과정 (Kruskal)

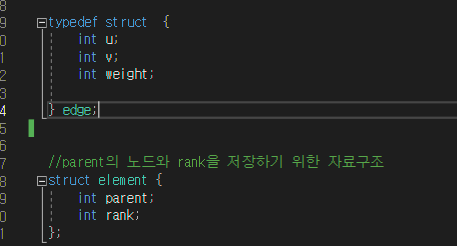
Kruskal:



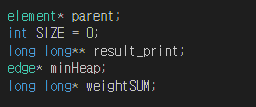
commands.txt 파일은 위처럼 저장했다.

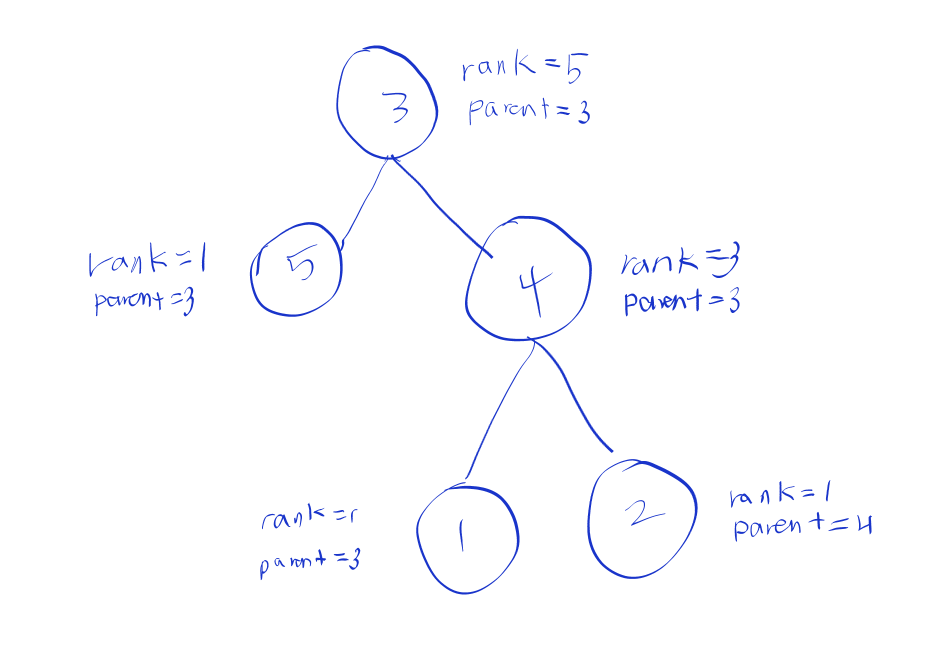
각 필요한 txt 파일은 main.cpp이 존재하는 directory 내부에 있다. 두번째 줄은 읽고 싶은 파일을, 마지막줄은 출력할 파일의 이름을 적는다. 이는 Prim도 동일하다.

자료구조)

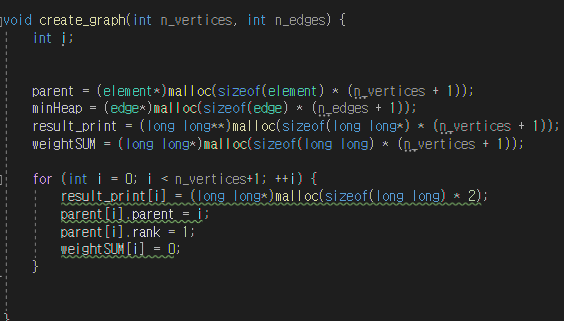


edge 자료를 저장할 edge 구조체와 노드의 parent와 노드의 개수를 세는 rank를 만들었다.

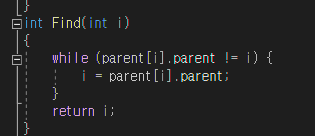
 프로그램을 실행하기 위해 만든 전역변수이다. parent는 노드의 부모를 저장한다. SIZE는 힙의 크기를 나타낸다. result\_print는 노드의 개수와 weight를 출력할 때 편리하게 하기 위해 만들었다. minHeap은 priority queue에서 필요한 heap 구조이다. weightSUM은 각 노드의 weight를 저장하는 자료구조이다. 아래 그림을 통해 구체적으로 나타내었다.

rank는 자신을 포함한 노드의 개수를 의미한다. root는 3이고 3의 rank는 5이다. leaf node의 rank는 1이고 parent는 노드 바로 위 노드를 의미한다.

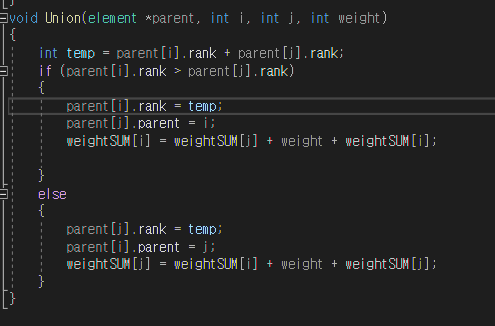
함수)



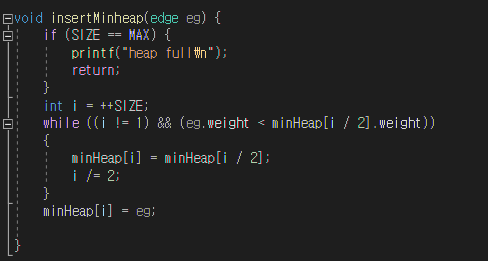
createGraph는 자료구조를 모두 할당하고 초기화하는 함수이다.



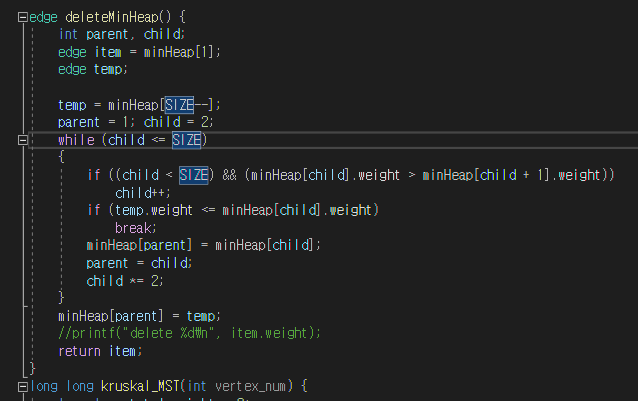
Find는 노드의 조상(root)를 찾는 함수이다.



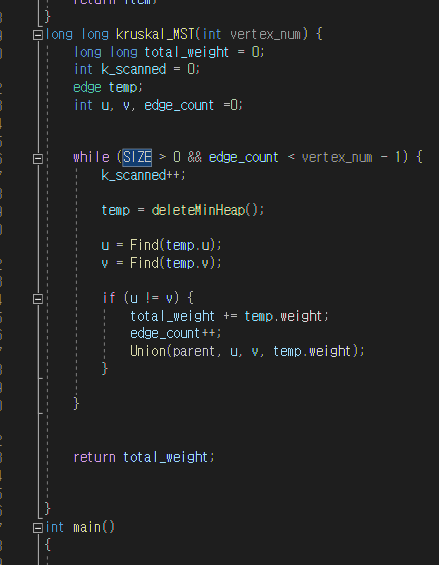
Union은 노드와 노드를 이어주는 함수이다. 하나의 노드의 rank가 다른 노드의 rank보다 높은 경우 부모로 선정하고 부모 rank는 이전 자신의 rank와 새로운 자식의 rank를 더한 값이 된다. 새로운 자식 노드의 parent는 다른 노드로 저장되고 weightSUM[부모] = weightSUM[부모] + weight[자식] + 새로운 edge의 weight 로 저장된다. 총 마지막 root의 weightSUM은 해당 트리의 total weight가 된다.



edge를 힙에 넣는 함수이다. 시간복잡도는 O(logN)이다.



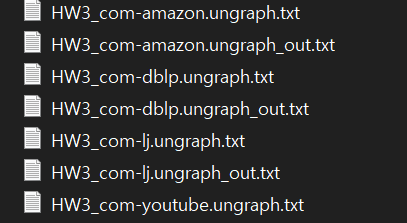
heap에서 가장 작은 weight를 가진 edge를 꺼내는 함수이다. 시간복잡도는 O(logN)이다.



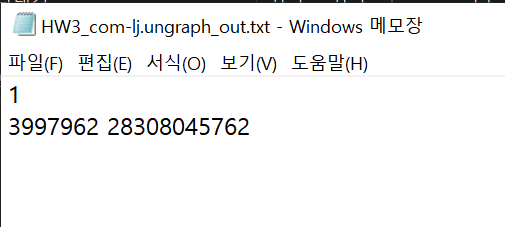
Kruskal 함수이다. Edge의 개수를 M, vertex 총 개수를 N이라 하면 총 시간 복잡도는 O(MlogN)이다. while은 최대 edge의 개수만큼 실행된다. 또한 while 내부에선 deleteMinHeap이 실행되고 이는 최대 vertex의 개수만큼 실행되기 때문에 O(MlogN)이 계산된다.

1. 실험 결과(Kruskal)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 파일 이름.txt | 작동여부 | mst weight | 수행시간(seconds) | kscanned |
| HW3\_email-Eu-core | yes | 3110161 | 0.05700 | 25571 |
| HW3\_com-dblp.ungraph | yes | 2747895457 | 1.57800 | 1049834 |
| HW3\_com-amazon.ungraph | yes | 2729670156 | 1.39000 | 925855 |
| HW3\_com-youtube.ungraph | yes | 14578691475 | 5.17800 | 2987623 |
| HW3\_wiki-topcats | yes | 5351181035 | 47.27700 | 28415895 |
| HW3\_com-lj.ungraph | yes | 28308045762 | 60.56100 | 34681165 |

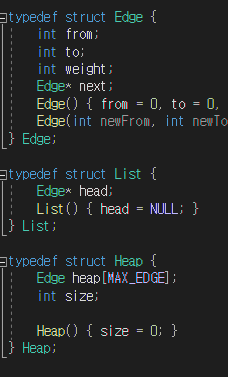


output 파일이 생성되고 output파일에는 weight들의 총 합이 저장되어 있다.



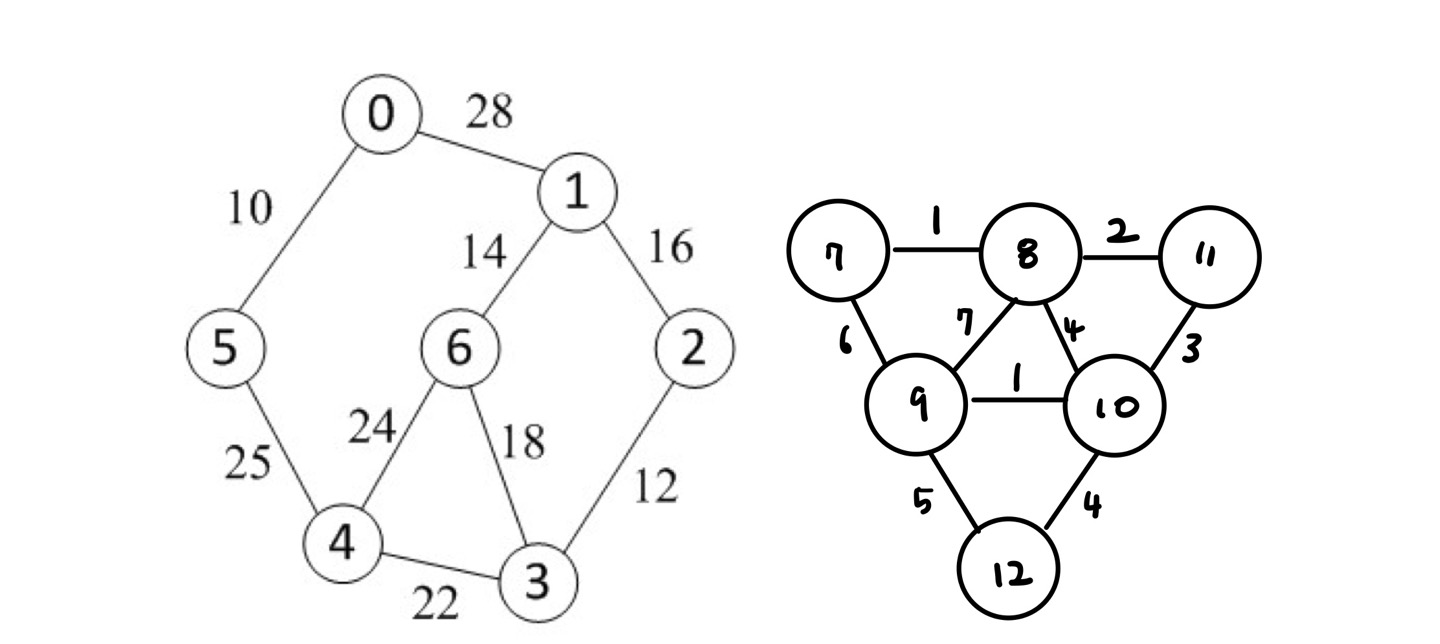
1. 실험 과정 (Prim)

자료구조)

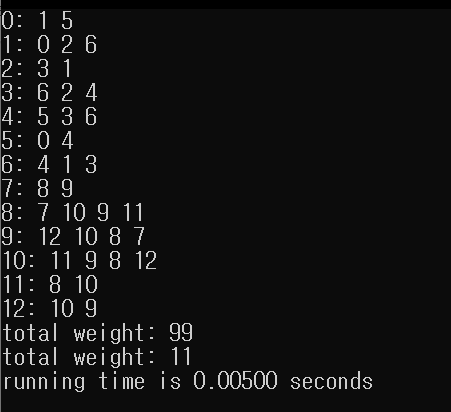


edge 구조체에는 출발 노드와 도착하는 노드, 그리고 해당 노드의 weight가 저장되어있다. Edge()를 통해 바로 초기화가 가능하고 Edge(from,to,weight)는 바로 원하는 값으로 초기화할 수 있는 구조체 함수이다.

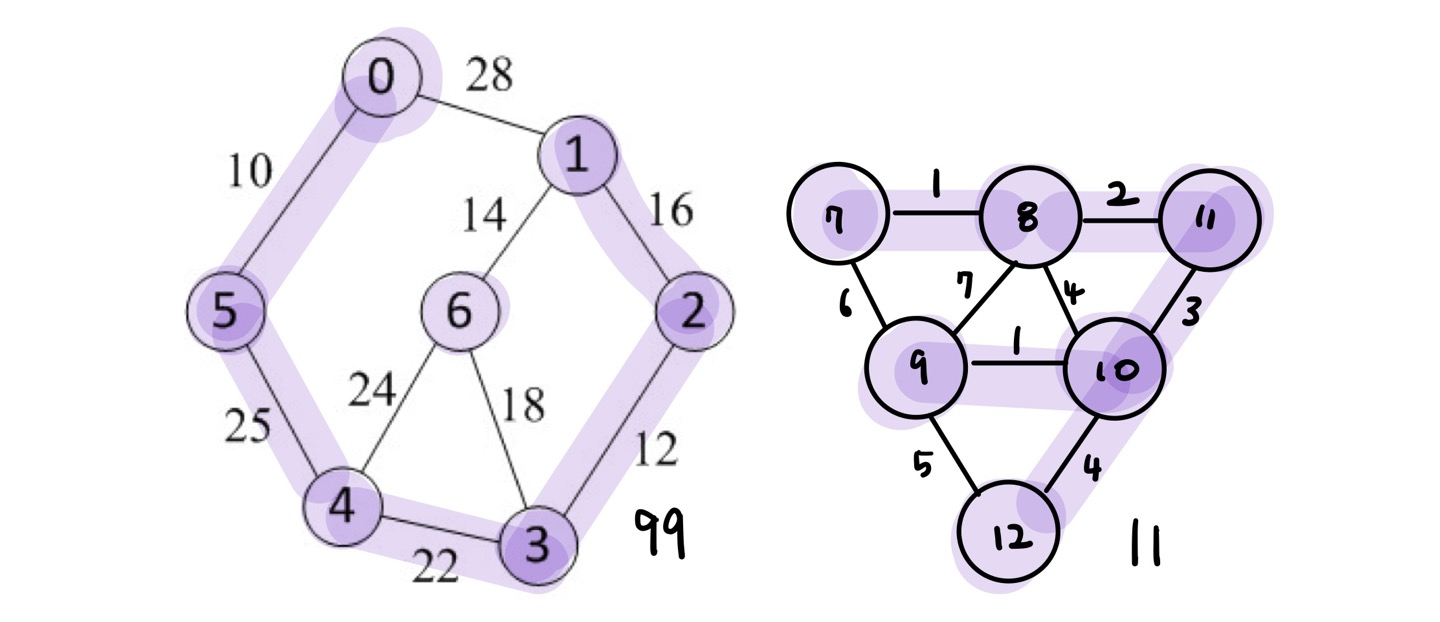
List는 Prim에서 필요한 인접 리스트를 저장하는 구조체이다. head는 해당 list의 시작을 의미한다.

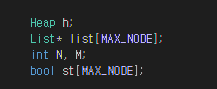


위와 같은 그래프가 존재한다고 하면



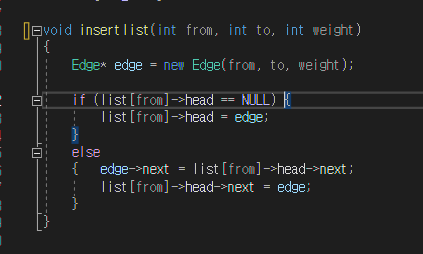
위 같은 list로 나타낼 수 있다. 0은 1과 5노드와 연결되어 있는데 출력된 값을 확인하면 list[0]에는 1과 5가 저장된 것을 확인할 수 있다. 또한 Prim MST 적용하면 첫번째 트리에서는 99, 두번째 트리에서는 11임이 출력된 것을 볼 수 있다. 그림으로 확인하면 다음과 같다.



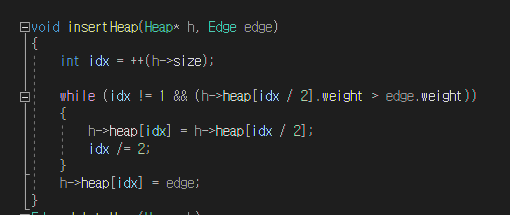


st는 해당 노드에 방문했는지 확인하는 함수이다. prim은 방문하지 않은 노드와 연결해야하기 때문에 해당 자료구조가 필요하다.

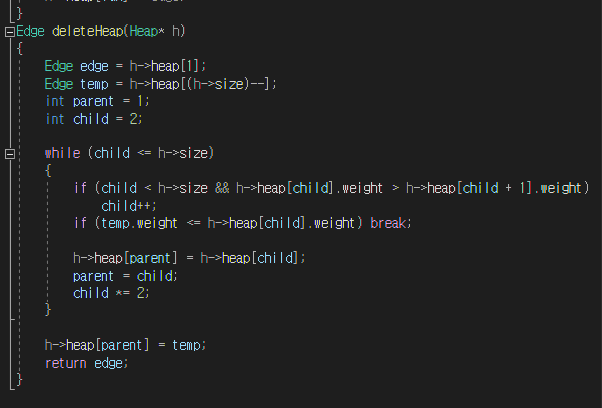
함수)



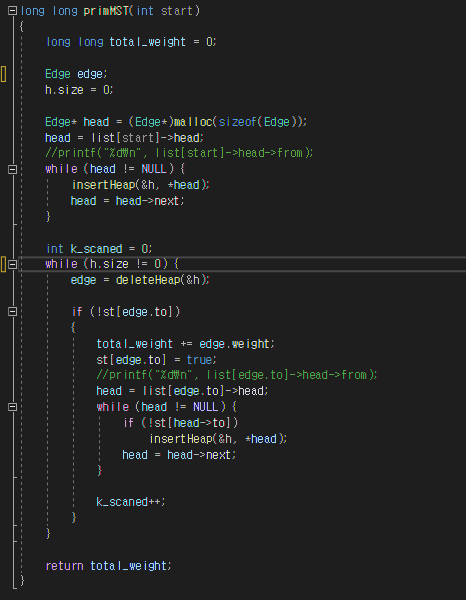
입력받은 edge를 list에 저장하는 함수이다. 아직 연결된 노드가 없으면 새로 연결하고, 아니면 맨 처음 노드로 지정한다. 즉 시간복잡도는 O(1)이다.



kruskal과 동일한 insertHeap함수이다. weight를 기준으로 minHeap을 구현했고 weight가 작을수록 높을 heap을 차지한다. 시간복잡도는 O(logN)이다.



deleteHeap은 가장 작은 weight를 가진 edge를 꺼내는 함수이다. 시간복잡도는 O(logN)이다.



primMST는 prim MST를 구현하는 함수이다. 해당 트리의 minimum spanning tree total weight를 반환한다. heap을 먼저 0으로 초기화한다. 그 뒤 MST에서 시작 노드의 head를 포인터로 지정한다. 이때 head가 NULL이 아닌 경우 시작 노드에 연결된 edge들을 모두 heap에 저장한다. K\_scaned=0으로 초기화하고 while문은 heap의 크기가 0이 아닐 때까지 실행한다. 이는 최대 edge의 개수만큼 실행된다. 또한 st는 방문하지 않으면 false이기 때문에 false일 때 다음 node로 넘어간다. st[다음노드]는 true로 업데이트 하고 다음 노드의 head에 접근해서 다음 노드에 연결된 edge들을 heap에 저장하는 알고리즘이다.

while은 최대 edge만큼 실행되고 insertHeap은 노드의 개수만큼 실행되기에 edge의 개수를 M, node의 개수를 N이라고 하면 O(MlogN)이 최종 시간복잡도가 된다.

1. 실험 결과 (Prim)

|  |  |  |  |
| --- | --- | --- | --- |
| 파일 이름.txt | 작동여부 | mst weight | 수행시간(seconds) |
| HW3\_email-Eu-core | yes | 3110161 | 0.03900 |
| HW3\_com-dblp.ungraph | yes | 2747895457 | 2.56400 |
| HW3\_com-amazon.ungraph | yes | 2729670156 | 2.18000 |
| HW3\_com-youtube.ungraph | yes | 14578691475 | 8.10800 |
| HW3\_wiki-topcats | yes | 5351181035 | 63.96300 |
| HW3\_com-lj.ungraph | yes | 28308045762 | 83.17700 |

1. TO 채점 조교님.

commands.txt는 main.cpp가 있는 디렉터리에 함께 있어야 합니다. 또한 실행하고 싶은 txt파일(ex HW3\_email-Eu-core.txt)도 main.cpp와 같은 디렉토리에 있어야 파일을 읽을 수 있습니다. 채점 수고가 정말 많으십니다. 한 학기동안 정말 감사했습니다.