1. 제목

정수기 검침원이 최단시간에 효율적으로 모든 가정에 방문하기 위한 MST 알고리즘 제안

* + what? 짧은 시간 안에 모든 가정에 방문하여 검침을 마치는 문제
  + why? 정수기 검침원의 효율적인 스케줄링를 위해
  + how? 방문 대상(정점) 간의 최소 신장 트리를 구성하는 알고리즘 연구

1. 문제 정의

한 동네를 돌며 본사의 정수기를 사용하는 가정에 방문하여 정수기 검침을 하는 사람이 있다. 신속하고 효율적인 일처리를 위해서는 검침을 빠른 시간에 끝낼 수 있도록 스케줄링을 해야한다.

1. 문제의 중요성

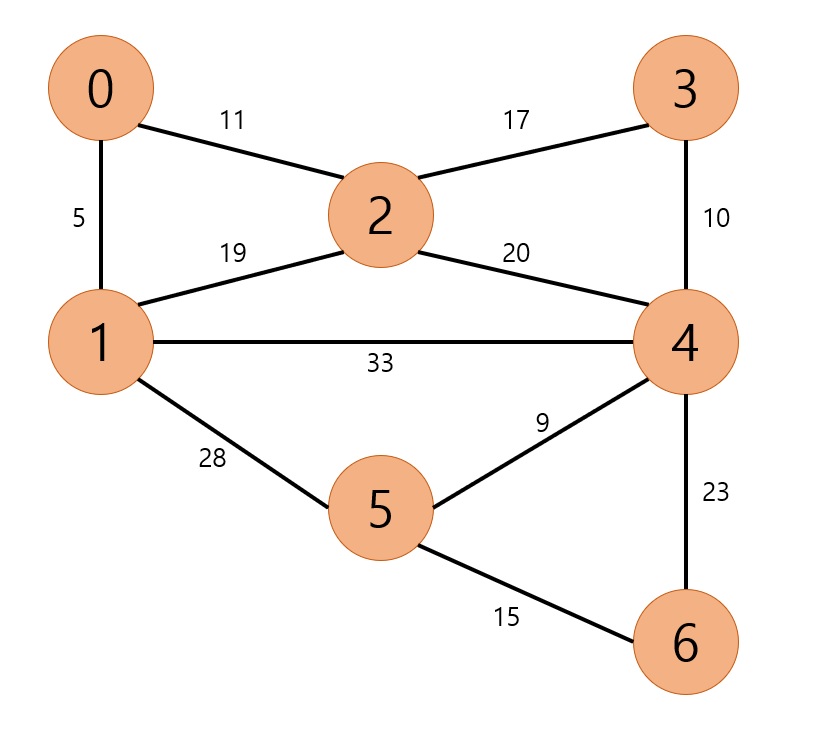
예를 들어 검침원의 한달 월급으로 생계를 꾸리기가 어려워 투잡을 뛰어야 한다고 가정하자. 낮 시간대에 최대한 빠르게 가정을 방문하여 정수기 검침을 마치는 것이 그에게 가장 중요한 숙제이다. 오후 5시 이전에 검침을 끝낸다면 집 앞 치킨집에서 주방 아르바이트를 할 수 있다. 따라서 각 가정별 이동 소요 시간을 최소화하는 것이 단시간에 효율적으로 일과를 끝낼 수 있는 방법이다.

1. 해법 제안

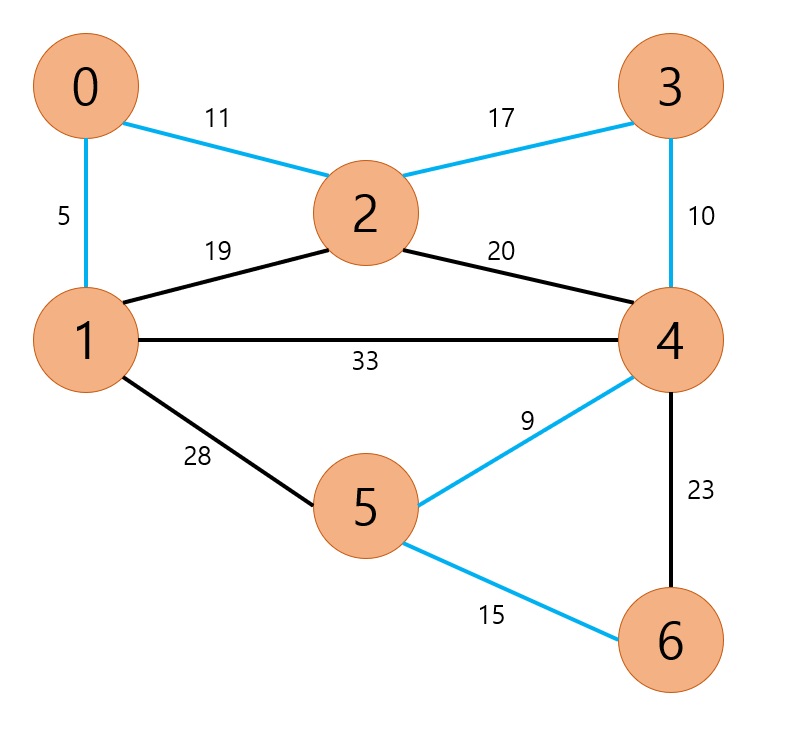
나는 이 문제를 해결하기 위해 'Kruskal의 최소비용 신장트리(MST) 알고리즘'을 사용할 것이다. 방문하고자 하는 장소(정점) 간의 거리는 항상 일정하게 주어져있고, 검침원은 모든 가정을 방문해야하기 때문에 가장 적은 시간이 걸리는 경로를 찾는 문제에 있어서 신장트리를 이용한 알고리즘을 적용하는 것이 알맞기 때문이다.

1. 실험 분석 결과

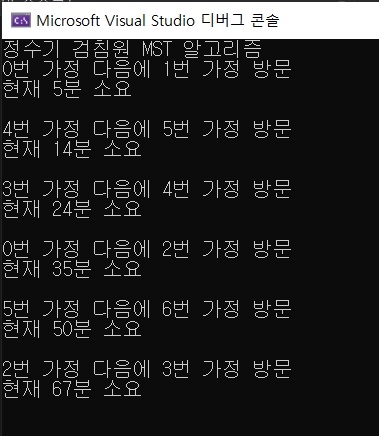
프로그램의 효율성을 검증하기 위해 다음 그래프 g를 입력값으로 준다고 가정하자.



* + g를 입력값으로 했을 때 예상되는 결과값은 최소신장트리인 아래 그래프이다.



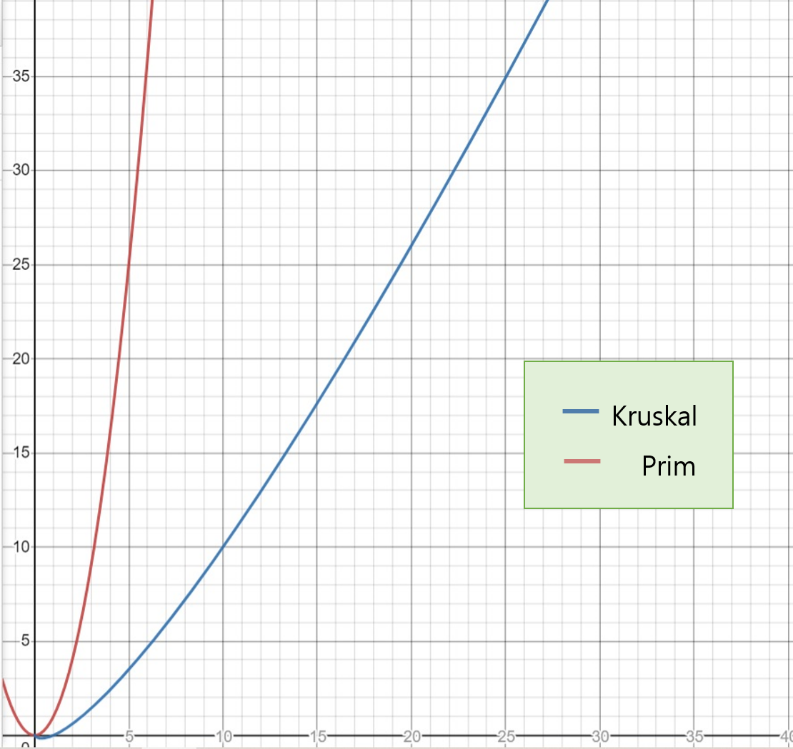
Kruskal 알고리즘을 구현한 프로그램에 주어진 그래프 g를 입력값으로 주었더니 다음과 같은 결과가 나왔다. 방문할 노드를 그래프 위에 표현해보면 예상한 결과값과 같으므로 Kruskal 알고리즘은 적절하다는 것이 증명되었다.



(시작지점이 설정되지 않았기 때문에 이 결과에서는 1번 가정을 시작지점으로 하여 1 > 0 > 2 > 3 > 4 > 5 > 6 순서로 방문한다고 보았다.)

* 이제 Kruskal 알고리즘이 Best Algorithm임을 증명하기 위해 동일한 결과값을 만들어주는 Prim 알고리즘과 비교를 실시해보자.

Kruskal 알고리즘과 Prim 알고리즘의 시간복잡도를 그래프 위에 나타내보았다.



정점의 개수 n과 간선의 개수 e에 대해서 Kruskal의 시간복잡도는 O(이고, Prim의 시간복잡도는 O(이다. 이에 따라 두 알고리즘의 장/단점을 비교해보자.

|  |  |  |
| --- | --- | --- |
|  | Kruskal | Prim |
| 시간복잡도 |  |  |
| 소스코드  라인 수 | 125 | 64 |

주어진 그래프 g에 대해서는 Kruskal 알고리즘이 시간복잡도가 더 작게 나왔으므로 효율적인 알고리즘이라고 할 수 있다. 그렇다면 n과 e값을 변경시키면 어떻게 될까?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | n = 5  e = 7 | n = 7  e = 13 | ... | n = 11  e = 28 | n = 15  e = 49 |
| Kruskal | 19.6 | 48.1 | ... | 134.4 | 274.4 |
| Prim | 25 | 49 | ... | 121 | 225 |

어느 한 지점에서부터 Prim이 더 효율적인 경우가 발생하게 된다. 하지만 내가 해결하고자 한 문제는 '정수기 검침원의 효율적인 가정 방문 경로' 문제이다. 그가 방문할 가정은 하루에 많아야 10개가 되지 않을 것이다. 따라서 희소그래프 문제로 바뀌게 된다. 희소그래프일 경우에는 Prim보다 Kruskal이 더 효율적이므로 코드가 길고 복잡하더라도 Kruskal 알고리즘이 적합한 해법이 될 수 있다.

1. 결론

위 분석에서 사용한 그래프에 따르면, 검침원은 1 > 0 > 2 > 3 > 4 > 5 > 6번 가정의 순서로 방문할 때 순수 이동 소요 시간이 67분으로 가장 짧은 시간 안에 모든 가정을 방문할 수 있다. 이 방법은 각 가정 간 거리만 파악할 수 있으면 모든 정점을 최단시간에 방문하는 그래프를 알려주기 때문에 유용하다. 하지만 간선의 개수(방문해야할 정점 사이의 노드 개수)가 많아질 경우 Prim 알고리즘을 사용하는 것이 효율적일 수 있다.

시간이 허락된다면, 그래프 정점과 간선의 개수를 검사해서 기준값보다 많으면 Prim 알고리즘을, 기준값보다 적으면 Kruskal 알고리즘을 사용하여 경로를 안내하는 프로그램을 만들어보고 싶다.

1. 참고문헌

이주영, 「최소신장트리를 위한 크루스칼 알고리즘의 효율적인 구현」, 『한국컴퓨터정보학회논문지』 제19권 제7호, 2014.07

오성현, 「최소신장트리를 이용한 다중고장 정전복구 방법」, 경남대학교 석사학위논문, 2015