

2020년 가을학기 알고리즘

Minimum Spanning Tree

데이터네트워크연구실
문현수, 이영석

munhyunsu@cs-cnu.org

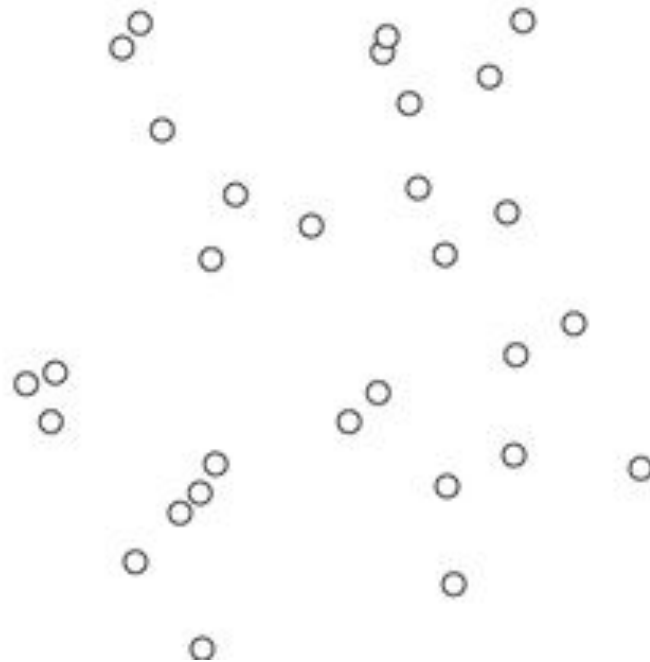
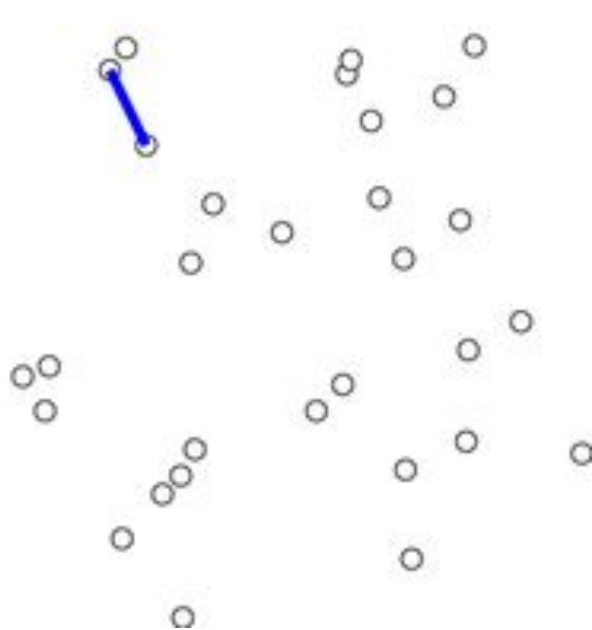
9주차 Feedback

- 제출률: 86%

이번주 실습 목표

- 최소 비용 신장 트리
 - Prim's Algorithm
 - Kruskal's Algorithm
- 실습1: Prim (2점)
- 실습2: Kruskal (3점)
- 실습3: 가장 중요한 길 (5점)

Prim vs. Kruskal



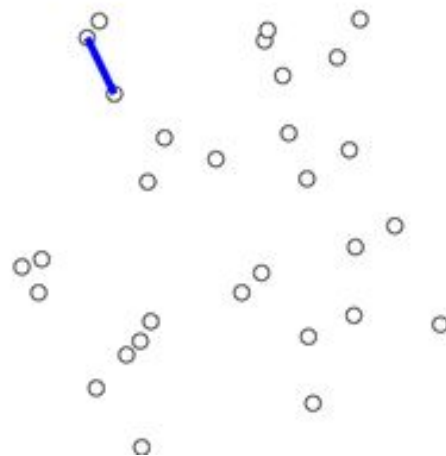
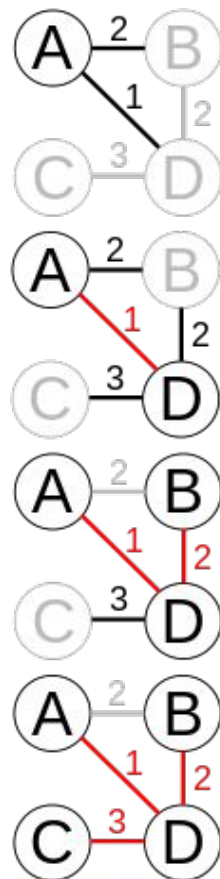
실습 1. Prim's Algorithm (2)

Prim's Algorithm

1. 그래프에서 하나의 노드를 선택하여 **트리**를 만든다.
2. 그래프의 모든 링크 집합을 만든다.
3. 모든 노드가 트리에 포함될 때까지
 - **트리**와 **연결된 링크** 중에서 트리 속의 두 노드를 연결하지 않는 링크 중 최소값의 링크를 트리에 추가한다.

The algorithm may informally be described as performing the following steps:

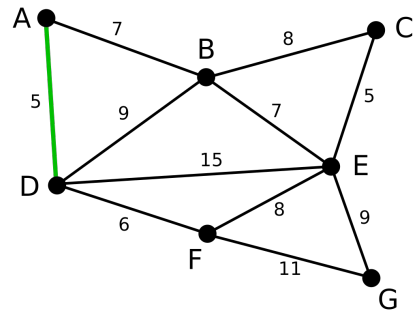
1. Initialize a tree with a single vertex, chosen arbitrarily from the graph.
2. Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
3. Repeat step 2 (until all vertices are in the tree).



A demo fo
based on I

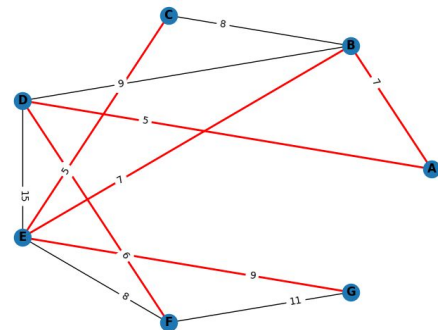
Prim's Algorithm 구현

- 무방향 가중치 그래프가 주어졌을 때 Prim 알고리즘으로 MST를 구하고, 그 에지의 합을 출력하시오.
만약 모든 노드가 연결되지 않아, MST가 없을 경우 0을 출력하시오.
 - 제한조건) Prim 알고리즘을 사용하지 않았을 경우,
코딩테스트 통과 결과에 상관없이 0점



```
[Input]
7 12
A B C D E F G
A B 7
C B 8
A D 5
C E 5
D B 9
E B 7
D E 15
E G 9
F G 11
E F 8
F D 6
E D 15
```

```
[Output]
39
```



```
F D 6
E D 15
harny@LuHa-X1C6 ~/Github/Algorithm/week10-mst $
cat 1.in | python3 prim.py
39
harny@LuHa-X1C6 ~/Github/Algorithm/week10-mst $
cat 1.in | java Prim
39
```

실습2. Kruskal's Algorithm (3)

Kruskal's Algorithm

algorithm Kruskal(G) **is**

```
F :=  $\emptyset$ 
for each  $v \in G.V$  do
  MAKE-SET( $v$ )
for each  $(u, v)$  in  $G.E$  ordered by  $\text{weight}(u, v)$ , increasing do
  if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ ) then
     $F := F \cup \{(u, v)\}$ 
    UNION(FIND-SET( $u$ ), FIND-SET( $v$ ))
return  $F$ 
```

function MakeSet(x) **is**

```
if  $x$  is not already in the forest then
   $x.\text{parent} := x$ 
   $x.\text{size} := 1$  // if nodes store size
   $x.\text{rank} := 0$  // if nodes store rank
end if
```

end function

function Find(x) **is**

```
if  $x.\text{parent} \neq x$  then
   $x.\text{parent} := \text{Find}(x.\text{parent})$ 
return  $x.\text{parent}$ 
```

else

return x

end if

end function



function Union(x, y) **is**

// Replace nodes by roots

$x := \text{Find}(x)$

$y := \text{Find}(y)$

if $x = y$ **then**

return *// x and y are already in the same set*

end if

// If necessary, rename variables to ensure that

// x has at least as many descendants as y

if $x.\text{size} < y.\text{size}$ **then**

$(x, y) := (y, x)$

end if

// Make x the new root

$y.\text{parent} := x$

// Update the size of x

$x.\text{size} := x.\text{size} + y.\text{size}$

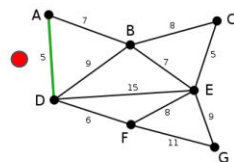
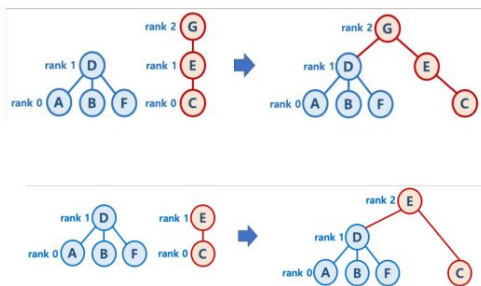
end function

참고) 이론 슬라이드

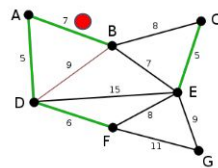
Union-by-rank

• Union-by-rank

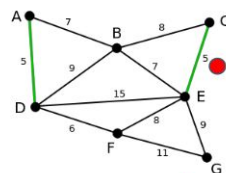
- 높이를 rank로 기억함
- 두 트리의 높이가 다르면
 - 높이가 작은 트리를 높이가 큰 트리에 붙임
- 두 트리의 높이가 같은 경우
 - 한 쪽 트리의 높이를 1 증가시켜 다른 트리의 루트 노드로 함



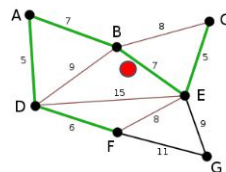
rank 1 D
↑
rank 0 A



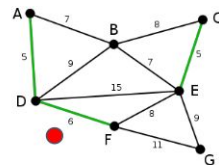
rank 1 D E
↗ ↑ ↖
rank 0 A B F C



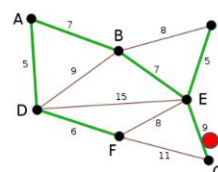
rank 1 D E
↑ ↑
rank 0 A C



rank 2 E
↗ ↖
rank 1 D C
↗ ↖
rank 0 A B F



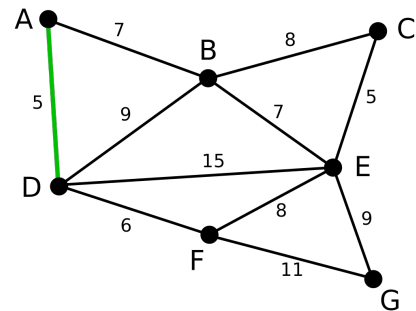
rank 1 D E
↑ ↖ ↑
rank 0 A F C



rank 2 E
↗ ↖
rank 1 D C G
↗ ↖
rank 0 A B F

Kruskal's Algorithm 구현

- 무방향 가중치 그래프가 주어졌을 때 **Kruskal** 알고리즘으로 **MST**를 구하고, 그 에지의 합을 출력하시오.
만약 모든 노드가 연결되지 않아도, 구해진 **MST** 에지 가중치 합을 출력하시오.
 - 제한조건) **Kruskal** 알고리즘을 사용하지 않았을 경우, 코딩테스트 통과 결과에 상관없이 0점

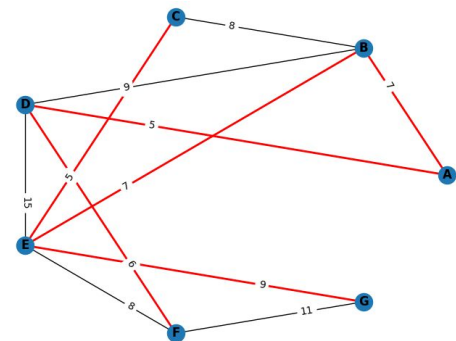


[Input]

```
7 12
A B C D E F G
A B 7
C B 8
A D 5
C E 5
D B 9
E B 7
D E 15
E G 9
F G 11
E F 8
F D 6
E D 15
```

[Output]

39



```
cat 1.in | python3 kruskal.py
39
harny@LuHa-X1C6 ~/Github/Algor
cat 1.in | java Kruskal
39
```

실습3. 가장 중요한 길 (5)

가장 중요한 길

- 규태는 네트워크 통신망 관리자다.
기관내 각 건물에 통신 설비가 설치되어
있고, 설비는 각기 다른 응답 속도를
보장한다.
통신에서는 응답 속도가 낮을 수록 좋다.
기관의 통신 장비와 링크 응답 속도가
주어졌을 때,
응답 속도 최소를 보장하기 위한 가장
중요한 길이 몇 개인지 출력하시오.

```
cat 3.in | python3 critical_path.py
2
harny@LuHa-X1C6 ~/Github/Algorithm/w
cat 3.in | java CriticalPath
2
```

[Input]

5 7

A B C D E

B A 1

A E 3

D A 2

B E 6

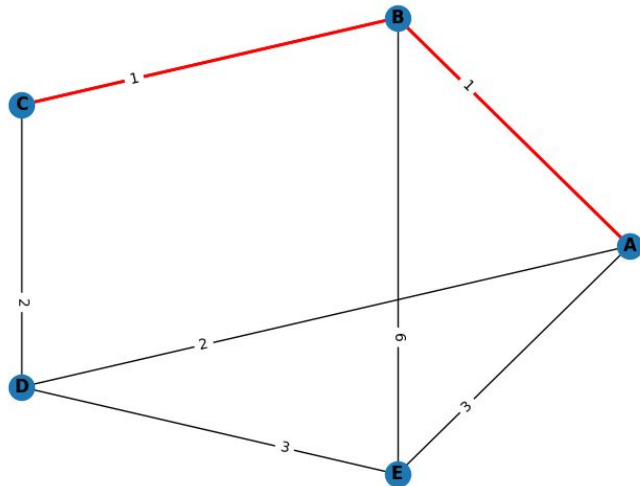
C B 1

C D 2

E D 3

[Output]

2



[설명]

가장 중요한 길: A-B, B-C

참고

Standard Library

- Python 3: <https://docs.python.org/3/library/>
- JAVA 11: <https://cr.openjdk.java.net/~iris/se/11/latestSpec/api/>
- C: <https://en.cppreference.com/w/c>
- CPP: <https://en.cppreference.com/w/cpp>

ID/ext↑↓	external ID↑↓	name↑↓
adb	ada	Ada
awk	awk	AWK
bash	bash	Bash shell
c	c	C
csharp	csharp	C#
cpp	cpp	C++
f95	f95	Fortran
hs	haskell	Haskell
java	java	Java
js	javascript	JavaScript
kt	kotlin	Kotlin
lua	lua	Lua
pas	pascal	Pascal
pl	pl	Perl
sh	sh	POSIX shell
plg	prolog	Prolog
py2	python2	Python 2
py3	python3	Python 3
r	r	R
rb	ruby	Ruby
scala	scala	Scala
swift	swift	Swift

주요 링크

- 코딩테스트:
<http://coding.cnu.ac.kr:8080/domjudge/public>
- FAQ :
<https://docs.google.com/document/d/1ntR6GS1SI7dRbYlw-pu8uT8U65Wc-RtMj10IEpiapfU/>
- 질문: 메일!
- 문의 사항: munhyunsu@cs-cnu.org
 - ID / PASSWORD 변경
 - 실습 시간외 질문: [AL]질문제목

잊지 않아야 할 것) 소스코드 및 보고서

- 사이버 캠퍼스에 목요일까지 제출
 - 추가 시간 필요한 학생들도 목요일까지 제출. 추가 시간 문제 해결은 메일로도 제출!
- 보고서(.pdf 파일), 소스코드(.java, .py 등) zip 파일 압축
 - AL_학번_이름_06.zip (메일 추가 제출) or AL19_06.zip (사이버캠퍼스)
- 시간/공간 복잡도 해석(STL 고려), 자신의 생각, 질문, 느낀점, 공유하고 싶은 문제
 - 문제 해결을 위해 어떤 접근법을 사용하였는지, 무엇을 배웠고 느꼈는가?

어디에 제출해야하는가? 헛갈린다.

- 제한 시간 내에 모두 해결했는가?
 - 네 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출
 - 아니요 -
 - 추가 시간 내에 해결한 문제가 있는가?
 - 네 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출하고, 메일로도 제출
 - 아니요 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출

보고서 템플릿: .pdf 로 제출!

- 알고리즘-**x**주차-주제
학번 이름

- 코드 테스트 결과 (점수표)

1	DOMjudge	0	0	
---	----------	---	---	--

- 각 문제별 내용

a. 문제 / 목표

b. 해결 방법 (주요 소스코드 첨부. 스크린샷 or 코드 CV)

c. 결과 (입력, 출력 결과)

- 느낀점: 과제를 하며 느낀 점 / 공유하고싶은 문제 / 난이도 / 부탁 / 조교에게...
등