



과 제 3

과목명	컴퓨터알고리즘
-----	---------

담당	천지영
----	-----

제출일	2020 / 05 / 0
-----	---------------

엘텍공과 대학/학부 사이버보안 학과

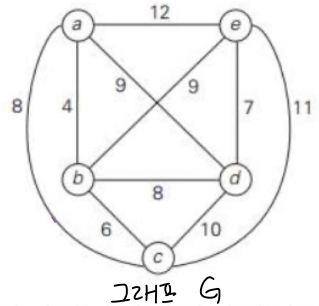
학 번	1771076
-----	---------

성 명	임은지
-----	-----



- 다음 그래프 G 에 대해 알고리즘 $\text{ApproxTSP}_1(G)$ 를 이용하여 근사 해를 구하고, 근사해를 구하는 과정을 알고리즘 단계별로 자세히 기술하시오.
- 위에서 구한 근사 해를 예로 들어 알고리즘 ApproxTSP_1 이 2-근사 알고리즘임을 확인해보시오.

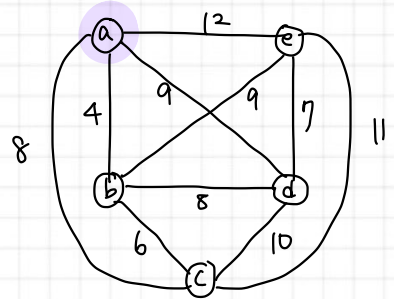
먼저, $\text{ApproxTSP}_1(G)$ 를 이용하여
근사해를 구한다.



1. Prim 알고리즘을 사용해서 그래프 G 의 최소 스패닝 트리 T 를 구한다.

① a 를 출발점으로 선택

인접 행렬	a	b	c	d	e
a	0	4	8	9	12
b	4	0	6	8	9
c	8	6	0	10	11
d	9	8	10	0	7
e	12	9	11	7	0



nearest

b	c	d	e
a	a	a	a

distance

b	c	d	e
4	8	9	12

→ Y 에 속한 점과의 최소 거리를
넣는다.

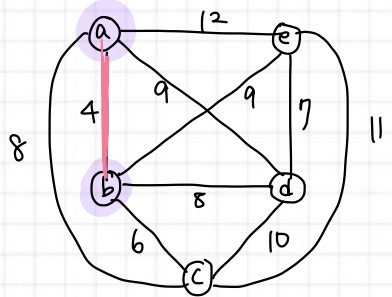
$Y = \{a\}$
(선택된 점의 집합 Y)

↳ Y 에 속한 정점 중 가장 가까운 정점을 넣는다.



- ② distance 가 가장 작은 b를 선택하여 nearest, distance 값을 갱신한다.

인접 행렬	a	b	c	d	e
a	0	4	8	9	12
b	4	0	6	8	9
c	8	6	0	10	11
d	9	8	10	0	7
e	12	9	11	7	0



$$Y = \{a, b\}$$

nearest

b	c	d	e
a	b	b	b

distance

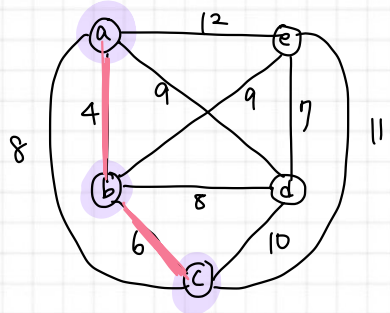
b	c	d	e
-1	6	8	9

↓ b→c, b→d, b→e
이미 선택한 것에는
-1 값을 넣는다.

$a \rightarrow c, a \rightarrow d, a \rightarrow e$ 보다
 $b \rightarrow c, b \rightarrow d, b \rightarrow e$ 가 각각
가중치가 더 작으므로, nearest
와 distance 값을 다음과
같이 채운다.

- ③ distance 가 가장 작은 c를 선택하여 nearest, distance 값을 ②에서 했던 방법으로 갱신한다.

인접 행렬	a	b	c	d	e
a	0	4	8	9	12
b	4	0	6	8	9
c	8	6	0	10	11
d	9	8	10	0	7
e	12	9	11	7	0



nearest

b	c	d	e
a	b	b	b

distance

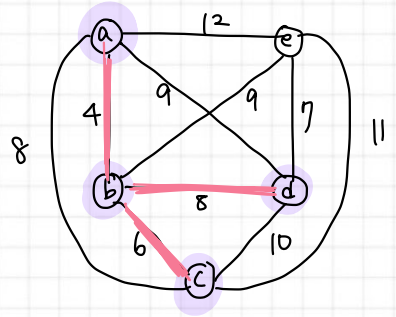
b	c	d	e
-1	-1	8	9

$$Y = \{a, b, c\}$$



④ distance 가 가장 작은 d 를 선택한다.

인접 행렬	a	b	c	d	e
a	0	4	8	9	12
b	4	0	6	8	9
c	8	6	0	10	11
d	9	8	10	0	7
e	12	9	11	7	0



nearest

b	c	d	e
a	b	b	d

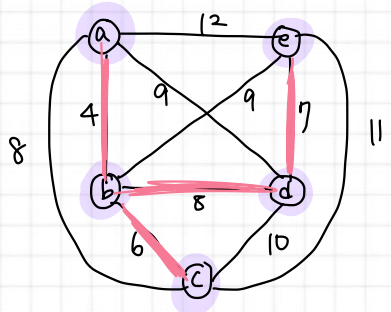
distance

b	c	d	e
-1	-1	-1	7

$$Y = \{a, b, c, d\}$$

⑤ distance 가 가장 작은 e 를 선택한다.

인접 행렬	a	b	c	d	e
a	0	4	8	9	12
b	4	0	6	8	9
c	8	6	0	10	11
d	9	8	10	0	7
e	12	9	11	7	0



nearest

b	c	d	e
a	b	b	d

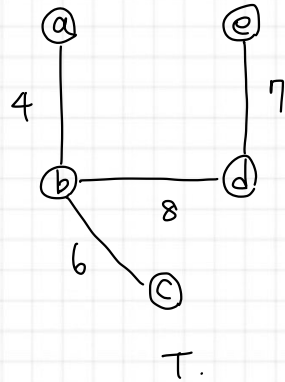
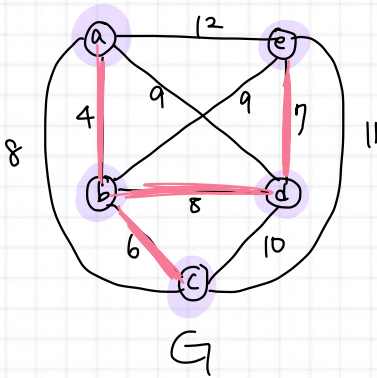
distance

b	c	d	e
-1	-1	-1	-1

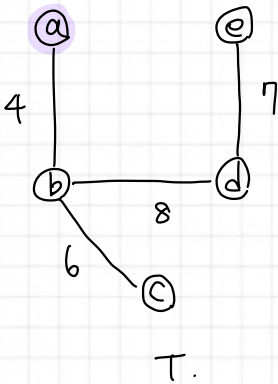
$$Y = \{a, b, c, d, e\}$$



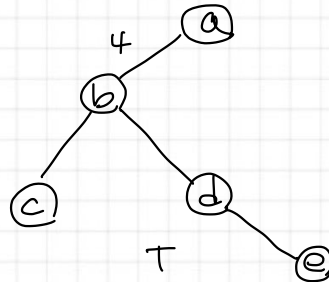
- ⑥ γ 에 G 의 모든 정점이 들어갔으므로, 그래프 G 의 최소 스패닝 트리 T 는 다음과 같다.



2. T 에서 임의의 정점 v 를 선택하여 루트로 한다.



a 를 선택하여 루트로 한다.
(보기 편하게 하기 위해 모양을 트리 모양으로 수정한다.)



3. v 가 루트인 T 를 중순위 순회하여 방문하는 정점들을 경로 r' 에 추가한다.

- ① a 가 루트인 T 를 중순위 순회한다.
 $a \rightarrow b \rightarrow c \rightarrow b \rightarrow d \rightarrow e \rightarrow d \rightarrow b \rightarrow a$
 순으로 방문하게 된다.



② 앞의 경로를 r' 에 추가한다.

$$r' = \{ a, b, c, b, d, e, d, b, a \}$$

4. r' 에서 각 정점이 처음으로 나오는 경로만 남겨두고 그 외에는 모두 제거한 경로를 r 이라 한다.

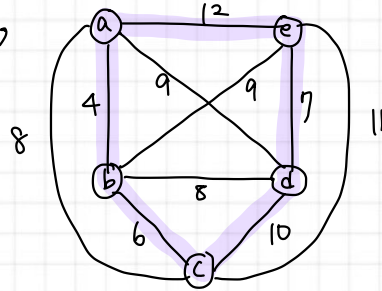
$$r = \{ a, b, c, d, e \}$$

5. r 이 사이클이 되도록 r 에 T 의 루트 v 를 마지막에 추가한다.

$$r = \{ a, b, c, d, e, a \} \text{ 이므로,}$$

그래프 G 에 대한 TSP 경로는

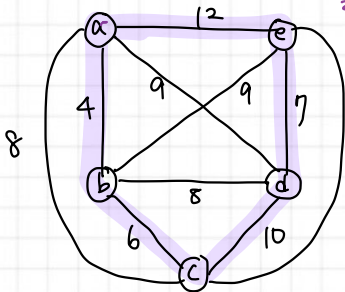
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow a$ 가
된다.



$$\therefore w(r) = 4 + 6 + 10 + 7 + 12 = 39$$

둘째로, $\text{ApproxTSP}_1(G)$ 알고리즘이 2- 근사 알고리즘임을 확인하기 위해 $w(r) \leq 2 \text{mst}$ 인지 확인한다.

① 확인하기 위해 그래프 G 의 TSP를 직접 구해본다.



출발점을 a 로 잡으면, a, b, c, d, e 를 한 번씩 거치고
 a 로 돌아와야 하므로, 앞의 a 와 뒤의 a 는 고정시키고,
가운데 b, c, d, e 를 사전적 배열로 순서를 바꿔가며
|| 거리를 계산한다. 이 그래프는 모든 정점이 자신을 제
외한 다른 모든 정점과 이어져있으므로, b, c, d, e 의 배열
 $4! = 24$ 개 만큼 cycle이 생긴다. 그런데,
 $a-b-c-d-e-a$ 와 $a-e-d-c-b-a$ 는 순서만 뒤집
어졌을 뿐 사용하는 간선은 같아 가중치가 같다. 따라서, 이렇게
접히는 경로를 제외한 12개 경로의 가중치만 살펴보면 된다.



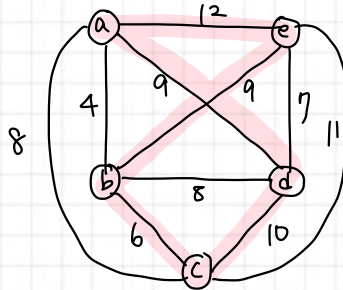
b c d e 의 사전식 배열

- | | | | |
|------------|------------------------|------------------------|------------------------|
| 1) b c d e | 7) c b d e | 13) d b c e | 19) e b c d |
| 2) b c e d | 8) c b e d | 14) d b e c | 20) e b d c |
| 3) b d c e | 9) c d b e | 15) d c b e | 21) e c b d |
| 4) b d e c | 10) c d e b | 16) d c e b | 22) e c d b |
| 5) b e c d | 11) c e b d | 17) d e b c | 23) e d b c |
| 6) b e d c | 12) c e d b | 18) d e c b | 24) e d c b |

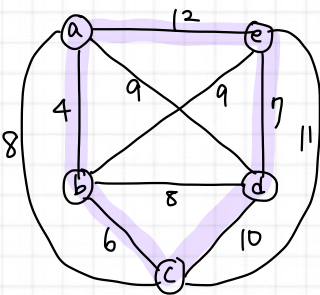
뒤집었을 때 겹치는 경로. ex) a d c b e a 는 같은

a e b c d a

간선을 사용한다.

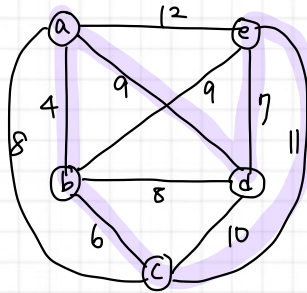


따라서 위의 12개의 경로에 대해 가중치를 구하면,



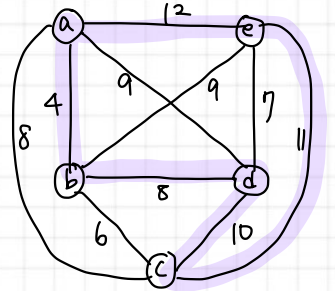
1) a-b-c-d-e-a

$$4+6+10+7+12=39$$



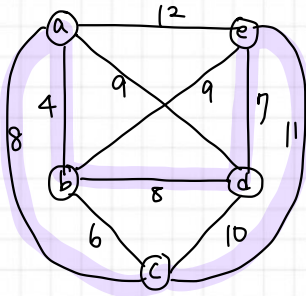
2) a-b-c-e-d-a

$$4+6+11+7+9=37$$



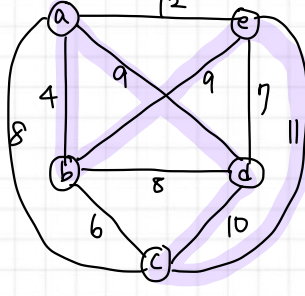
3) a-b-d-c-e-a

$$4+8+10+11+12=45$$



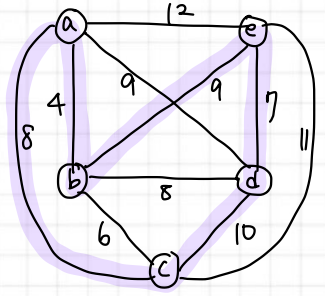
4) a-b-d-e-c-a

$$4+8+7+11+8=38$$



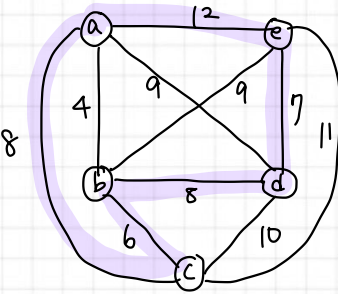
5) a-b-e-c-d-a

$$4+9+11+10+9=43$$



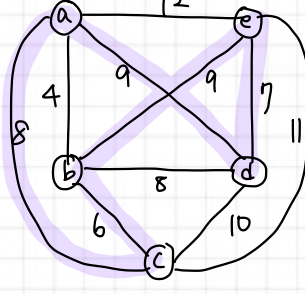
6) a-b-e-d-c-a

$$4+9+7+10+8=38$$



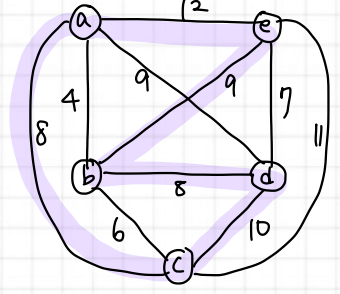
7) a-c-b-d-e-a

$$8+6+8+7+12=41$$



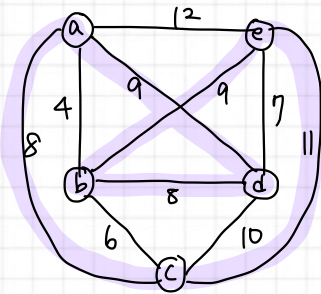
8) a-c-b-e-d-a

$$8+6+9+7+9=39$$



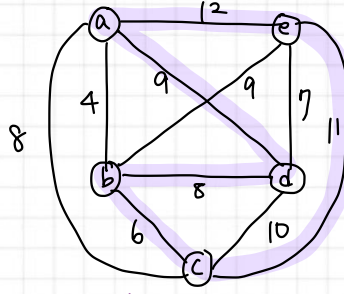
9) a-c-d-b-e-a

$$8+10+8+9+12=47$$



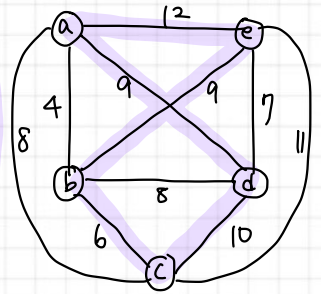
11) a-c-e-b-d-a

$$8+11+9+8+9=45$$



13) a-d-b-c-e-a

$$9+8+6+11+12=46$$



15) a-d-c-b-e-a

$$9+10+6+9+12=46$$

$\therefore \text{mst} = 39$ (when, a-b-c-e-d-a)



$\therefore w(r) = 39$ 이고, $2 \times \text{mst} = 2 \times 37 = 74$ 이므로,

$w(r) < 2\text{mst}$ 이다.

따라서, ApproxTSP1 알고리즘은 2-근사 알고리즘이라
할 수 있다.