CS217 - Algorithm Design and Analyis
Homework Assignment 3

Group Name: Static
Group Members: Zhenjia Xu, JiaCheng Yang
Zhuolin Yang, Wenda Qiu

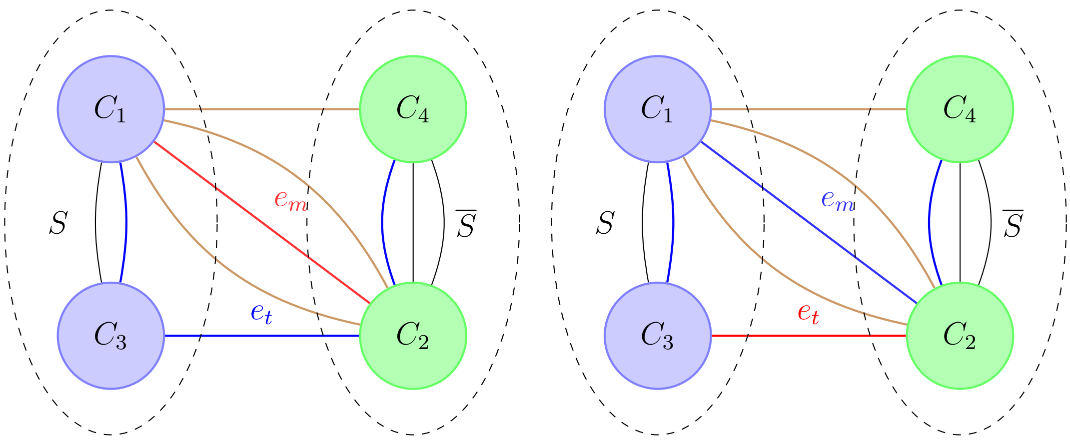
31st Oct, 2016

3 Minimum Spanning Trees

Exercise 3.1.

Lemma 1 (**Cut Lemma**)**.** *Let $X \subseteq E$ be a good set. If (V, X) is not a spanning tree, then (V, X) consists of two or more connected components. Let $V = S \cup \overline{S}$ be a cut of X. That is, no edge of X goes from S to $\overline{S}$. Let $e \in E$ be an edge of minimum cost connecting two connecting components of (V, X). Then $X \cup \{e\}$ is good, too.*

Proof. Suppose the $X \cup \{e\}$ is not a good set. Then all the spanning trees exclude the minimum edge which connects two connecting components of (V, X). Denote these two connecting components as C_1 and C_2. Pick up one of the minimum spanning trees T_m and substitute the edge $e_t \in E(T_m)$ connecting C_1 and C_2 to e_m.



(a) Original minimum spanning tree (b) Generated spanning tree

Figure 1: The illustration of the proof

Then the "costs" of the new spanning tree T'_m is:

$$c(T'_m) = \sum_{e \in E(T')} c(e) = c(T_m) - c(e_t) + c(e_m)$$

Using the condition e_m is the minimum edge connecting C_1 and C_2, we can easily know that:

$$c(e_m) \leq c(e_t)$$

Then we obtain a more optimal spanning tree(or we obtain another minimum spanning tree containing the edge e_m) which leads the contradictory according to the inequality below:

$$c(T'_m) = c(T_m) - c(e_t) + c(e_m) \leq c(T_m)$$

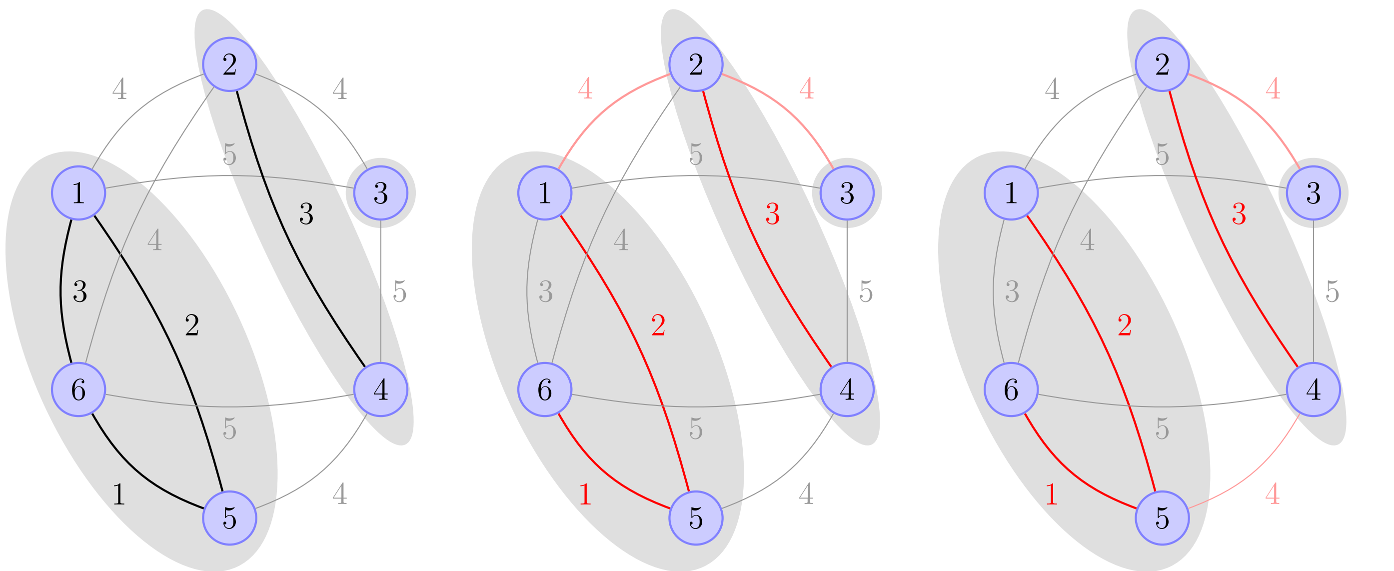
Hence, $X \cup \{e\}$ is good. $\square$

Lemma 2 (**The Inverse of Cut Lemma**). *If X is good, $e \notin X$, and $X \cup \{e\}$ is good, then there is a cut S, $V \backslash S$ such that the following two holds:*

1. *no edge from X crosses this cut;*
2. *e is minimum weight edge of G crossing the cut.*

Proof. TODO $\square$

Exercise 3.4. The example is illustrated below, in which we can see that no matter which minimum spanning trees we choose, the connected components is indentical.



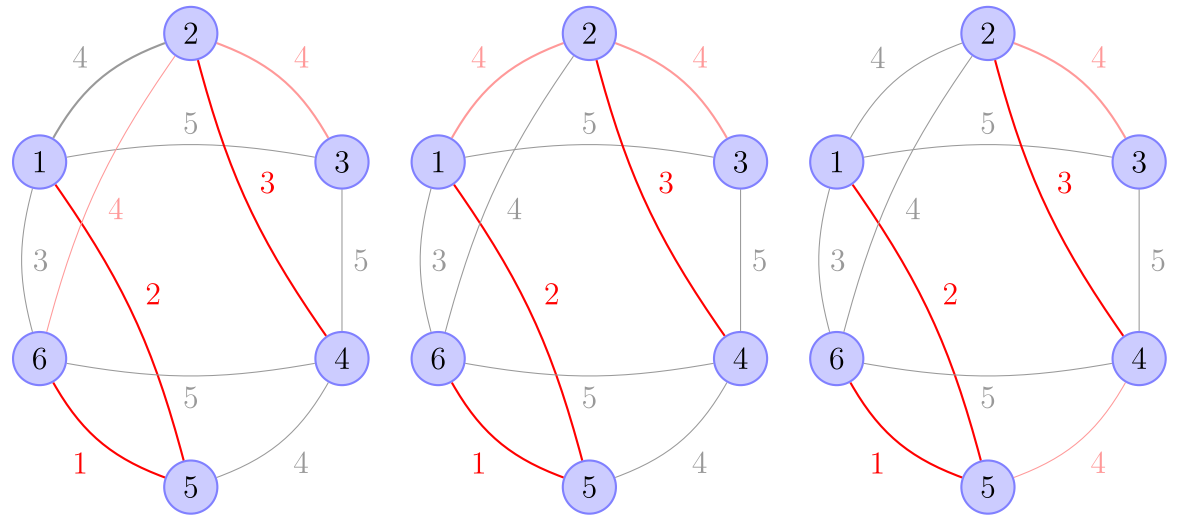
(a) The original graph and its connect components only considering the edges whose weight is less than 4.

(b) A possible minimum spanning tree and its connect components only considering the edges whose weight is less than 4.

(c) Another possible minimum spanning tree and its connect components only considering the edges whose weight is less than 4.

Figure 2: The illustration of the proof

Exercise 3.7. The example is illustrated below, in which we can see that no matter which minimum spanning trees we choose, the m_3 of each minimum spanning trees is indentical. The edges colored red are the minimum spanning tree.



(a) A possible minimum spanning tree

(b) Another possible version of the minimum spanning tree

(c) The remained possible minimum spanning tree

Figure 3: Three possible minimum spanning trees and its m_3

Exercise 3.8. Suppose that there are two different minimum spanning tree, called T, T'.

We can find the edge which belongs to only one of the minimum spanning tree and its weight is minimal. Because no two edges of G have the same weight, the above edge is unique, denoted as e. (Without loss of generality, let e belong to T)

By Lemma 3, we known that

$$m_{c_e}\{T\} = m_{c_e}\{T'\} \text{ and } m_{c_e-1}\{T\} = m_{c_e-1}\{T'\}$$

So

$$m_{c_e}\{T\} - m_{c_e-1}\{T\} = m_{c_e}\{T'\} - m_{c_e-1}\{T'\}$$

However, e belongs to only T, in other word

$$m_{c_e}\{T\} - m_{c_e-1}\{T\} = 1 \text{ and } m_{c_e}\{T'\} - m_{c_e-1}\{T'\} = 0$$

which contradicts the equation above.

So, the hypothesis is wrong; that is to say G has only one minimum spanning tree.

Exercise 3.10. The two parts of the graph are independent, so we only need to caltulate the minimal spanning tree of the two parts respectively.

- Left part:

  If we don't choose any of the parallel edges(containing three edges), we need to choose the other two edges.(1 choice).

  If we choose one of the parallel edges(3 choices), we only need to choose one edge from the other two edges(2 choices).

  In summary, we have $1 + 3 \times 2 = 7$ minimal spanning trees in the left part.

- Right part:

  Using the similar method, the right part have $1 + 2 \times 3 = 7$ minimal spanning trees in the right part.

- Combining two parts:

  Using Multiplication rule, we can get the total amount of the minimal spanning forests, namely $7 \times 7 = 49$

We also justify the answer by programming(enumerating every set of the edges, and check if it can form a minimal spanning forest):

```
u = [1, 1, 2, 2, 2, 4, 4, 5, 6, 4] # Edge set
v = [2, 3, 3, 3, 3, 5, 5, 6, 7, 7]

def getfa(x) :
    if (f[x] == x) :
        return x
    else:
        f[x] = getfa(f[x])
        return f[x]

if __name__ == '__main__' :
    answer = 0
    for mask in range(1, 1 << 10) : # Enumerate every possible edge sets.
        flag = 1
        number = 0
        f = [i for i in range(8)]
        for i in range(0, 10) :
            if (((mask >> i) & 1) == 1) :
                number = number + 1
                if (getfa(u[i]) == getfa(v[i])) :
                    flag = 0
                else:
                    f[getfa(u[i])] = getfa(v[i])
        if (number == 5 and flag) :
            answer = answer + 1
    print answer
```