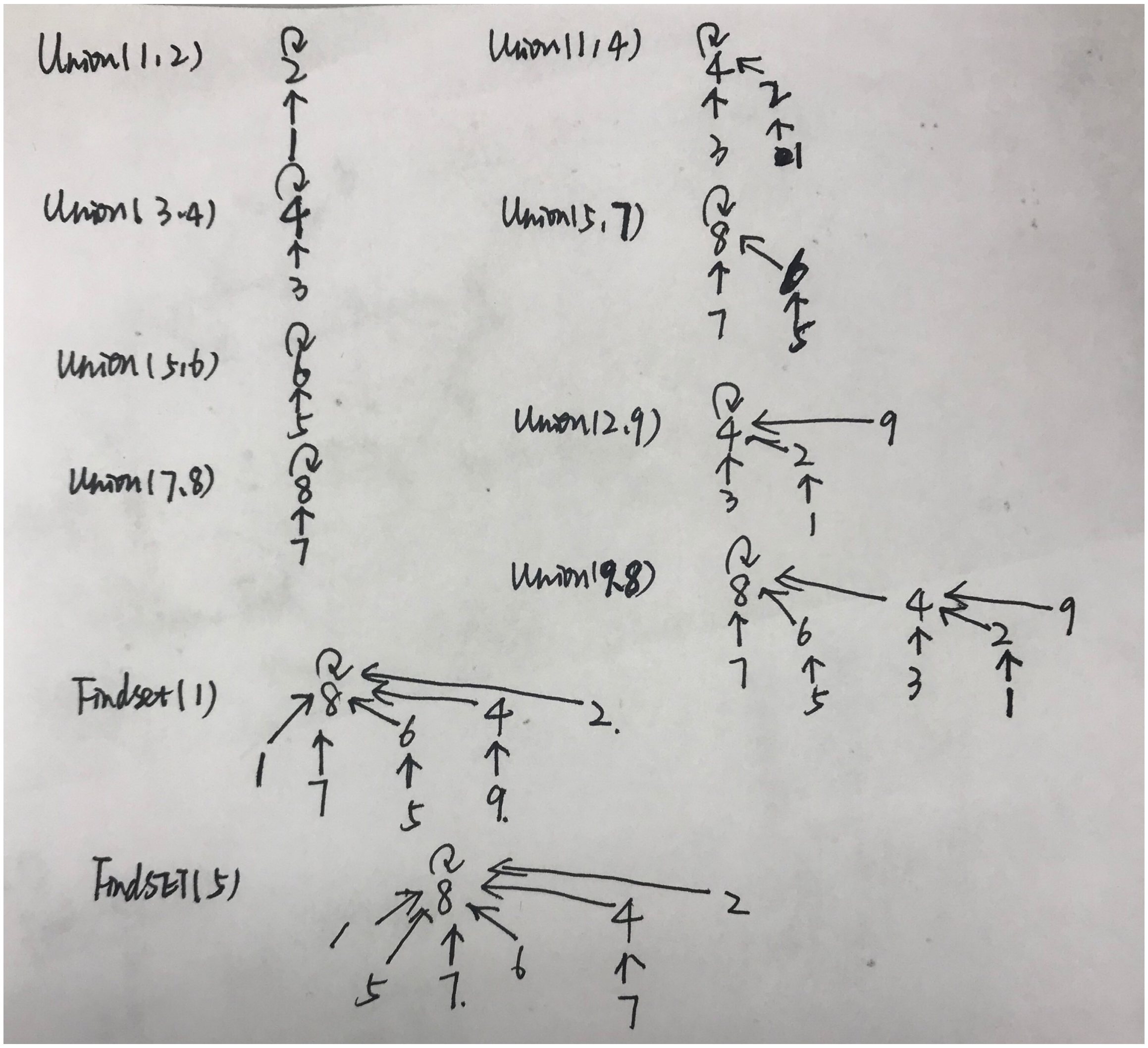
1.(a) The cost of a minimum spanning tree is 23.

(b)There are two minimum spanning trees.

2.

V	$v.key$	$v.\pi$
a	$\infty \to 0$	nil
b	$\infty \to 13 \to 7 \to 6$	nil $\to$ a $\to$ d $\to$ h
c	$\infty \to 1$	nil $\to$ a
d	$\infty \to 3$	nil $\to$ c
e	$\infty \to 2$	nil $\to$ a
f	$\infty \to 9 \to 3$	nil $\to$ e $\to$ d
g	$\infty \to 5 \to 4$	nil $\to$ e $\to$ f
h	$\infty \to 4$	nil $\to$ a

3.



4.

m	1	2	3	4	5	6
1	0	315	1782	1827	1752	5127
2		0	567	1377	1527	2427
3			0	1890	2240	4340
4				0	1350	9450
5					0	3000
6						0

s	1	2	3	4	5	6
1		1	1	1	1	1

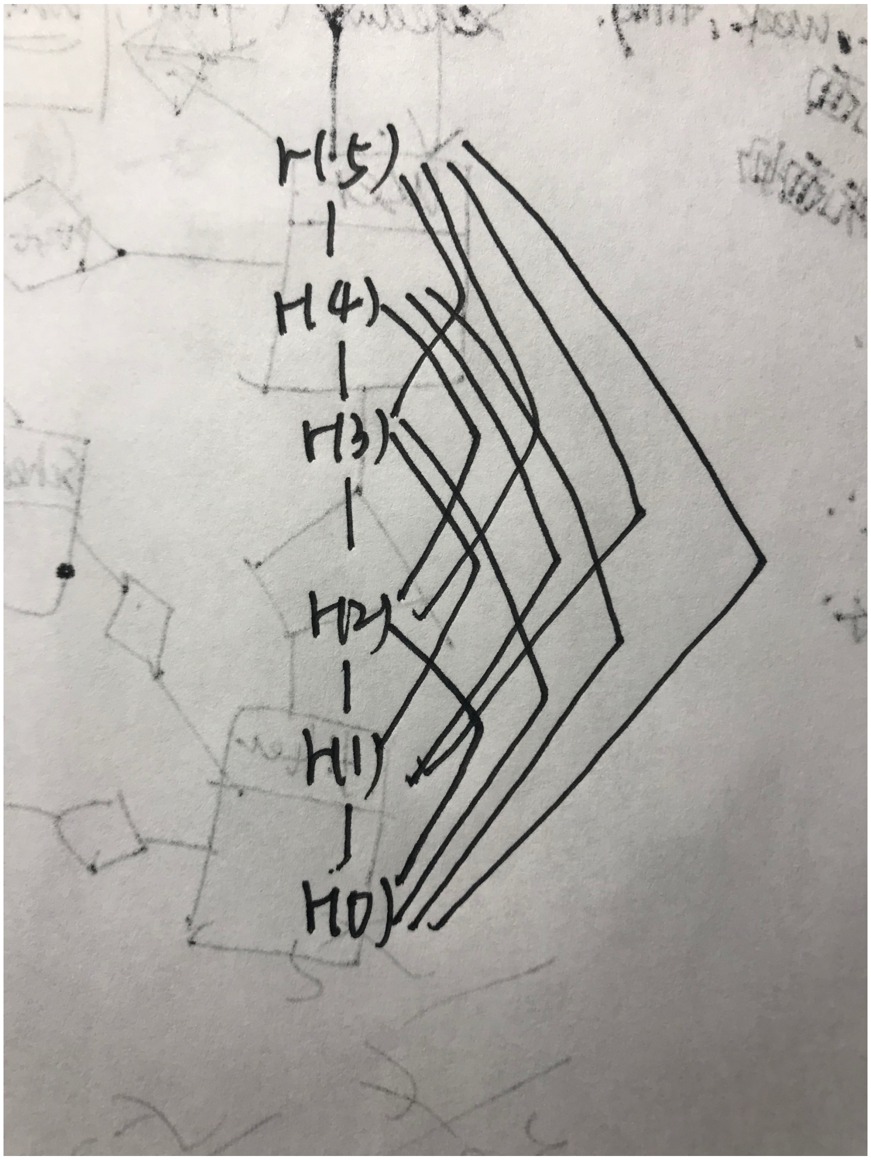
2			2	3	4	5
3				3	4	5
4					4	5
5						5
6						

Therefore, the best way to multiply these matrices is $(A_1\left(A_2\left(A_3\left(A_4(A_5A_6)\right)\right)\right))$.

5. Her claim is not correct. There is a counter example to prove this. Consider the following case A1 is 5*6, A2 is 6*3, A3 is 3*2. According to her claim, then the result is (A1A2)A3=120. But we can group in another way (A1(A2A3)=96 which is smaller than previous result, so her claim is not right.

6.The recurrence formula is $r_n = \max_{1\le i\le n}\{P_i - c + r_{n-i}\}$

```
def   Cut_Rod(p,n)
      let r[0..n] be a new array
      r[0]=0
      for j=1 to n:
            q=-∞
            for i=1 to j
                  if(i!=j)
                        q=max(q,p[i]+r[j-i]-c)
                  else
                        q=max(q,p[i])
            r[j]=q
return r[n]
```



1	6	8	11	13

7.Firstly, I add the edge which is reduced to T, and then there exists a circle. Secondly, I take advantage of depth_first search to find the maximum edge in the circle and delete it. The running time is O(n+m) because dfs's running time is O(n+m).

8.Firstly, turn bandwidth of every edge into its opposite which is a negative number. And then using the MST algorithm in the lecture to find new graph's MST, which is the maximum spanning tree of the original tree. Finally, cut part of the path starting from a to b. And this is the answer. The running time is O(ElogV).

9.Recurrence formula:

$$f(i)=\min_{m_i-m_j<300 \wedge 0<j<i}(h_i+f(j))$$

```
def Hotel_Cost(m,h,n)
    let c[0..n] be a new array
    c[0]=0
    m[0]=0
    for j=1 to n
        q=∞
        i=j-1
        while(m[j]-m[i]<300&&i>=0)
            q=min(q, h[j]+c[i])
            i--
        c[j]=q;
    return c[n]
```