

Ques 1:

a) Solve the recurrence relation

Masters theorem

$$T(n) = aT\left(\frac{n}{b}\right) + n^k \log^p n$$

$a \geq 1$, $b > 1$, $k \geq 0$, p real no.

1) if $a > b^k$ then $T(n) = \Theta(n^{\log_b a})$

2) if $a = b^k$

a) if $p > -1$ then $T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$

b) if $p = -1$ then $T(n) = \Theta(n^{\log_b a} \log \log n)$

c) if $p < -1$ then $T(n) = \Theta(n^{\log_b a})$

3) if $a < b^k$

a) if $p \geq 0$ then $T(n) = \Theta(n^k \log^p n)$

b) if $p < 0$ then $T(n) = O(n^k)$

i). $T(n) = 6T\left(\frac{n}{3}\right) + n^2 \log n$

$a = 6$ $b = 3$ $k = 2$ $p = 1$

Here, $a < b^k$ $6 < 3^2$ $6 < 9$

$$p=1 \quad \text{i.e. } p > 0$$

$$T(n) = \Theta(n^2 \log n)$$

ii). $T(n) = 2T(\sqrt{n}) + \log n$

$$\text{let } n = 2^m \Rightarrow \log n = m$$

$$T(2^m) = 2T(2^{m/2}) + m$$

$$\text{let } T(2^m) = S(m)$$

$$S(m) = 2S(m/2) + m$$

Using Masters Theorem,

$$a=2 \quad b=2 \quad k=1 \quad p=0$$

$$a = b^k \quad \text{and } p > -1$$

$$S(m) = \Theta(m^{\log_b a} \log^{p+1} m) = \Theta(m^{\log_2 2} \log^1 m) = \Theta(m \log m)$$

$$T(2^m) = \Theta(m \log m)$$

$$T(n) = \Theta(\log n \log \log n)$$

b). Time Complexity of program

void fun (n)

{

int i, j, count = 0 ;

for (i=n ; i>0 ; i=i/2)

for (j=0 ; j<=i ; j++)

count ++ ;

}

Since j loop is dependent on i, so lets unroll it

Value of i	n	n/2	n/4	n/8	n/n
j loop runs	n	n/2	n/4	n/8	n/n

Total j loop runs: $n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \dots + \frac{n}{n}$

$$: n \left(1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^{\log n}} \right)$$

Sum of GP

$$= n \left(\frac{1}{1 - \frac{1}{2}} \right) = 2n = O(n)$$

Ques 2:

Logic: Divide the array in 2 parts and compare the maximum and minimum of the 2 parts to get the maximum and minimum of whole array

```
#include <iostream>

using namespace std;

struct Pair {
    int min;
    int max;
};

struct Pair fun(int arr[], int low, int high)
{
    struct Pair sp ;

    // If there is only one element
    if (low == high)
    {
        sp.max = arr[low];
        sp.min = arr[low];
        return sp;
    }
}
```

```

// If there are two elements
if (high == low + 1)
{
    if (arr[low] > arr[high])
    {
        sp.max = arr[low];
        sp.min = arr[high];
    }
    else
    {
        sp.max = arr[high];
        sp.min = arr[low];
    }
    return sp;
}

// If there are more than 2 elements
int mid = (low + high) / 2;
struct Pair lp = fun(arr, low, mid);
struct Pair rp = fun(arr, mid + 1, high);

// Compare minimums of two parts
if (lp.min < rp.min)
    sp.min = lp.min;
else
    sp.min = rp.min;

// Compare maximums of two parts
if (lp.max > rp.max)
    sp.max = lp.max;
else
    sp.max = rp.max;

return sp;
}

int main()
{
    int arr[] = {100, 11, 35, 8, 55, 30};
    int n = sizeof(arr)/sizeof(int);

    struct Pair res = fun(arr, 0, n - 1);

    cout << "Minimum element is " << res.min << endl ;
    cout << "Maximum element is " << res.max << endl ;

    return 0;
}

```

Recurrence Relation : $T(n) = 2T\left(\frac{n}{2}\right) + 1$

Solving: $a=2$ $b=2$ $k=0$ $p=0$

$$a > b^k$$
$$2 > 2^0$$

$$T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$$

Ques 3:

```
#include <iostream>

using namespace std;

int find(int arr[], int size)
{
    int low = 0;
    int high = size-1;

    while (low <= high)
    {
        int mid = (low + high)/2;

        if (arr[mid] < mid + 1)
            low = mid + 1;
        else if (arr[mid] > mid + 1)
            high = mid - 1;
        else
            return mid ;
    }
    return -1;
}

int main()
{
    int arr[] = {0,1,3,5,6,7};
    int n = sizeof(arr)/sizeof(int);
    cout << find(arr, n) ;
    return 0 ;
}
```

arr: $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ \begin{bmatrix} 0 & 1 & 3 & 5 & 6 & 7 \end{bmatrix} \end{matrix}$

Ans: 2 index

Ques 4:

Assumption: Every job takes 1 unit time for completion.

a).

jobs	J1	J2	J3	J4	J5	J6
Deadlines	5	3	2	2	4	1
Profits	200	180	190	300	120	100

- Sort the jobs in decreasing order of profit

jobs	J4	J1	J3	J2	J5	J6
Deadlines	2	5	2	3	4	1
Profits	300	200	190	180	120	100

- Iterate over the jobs and assign the last slot available
Profit

0	1	2	3	4	5	6

0

J4:

	J4					
0	1	2	3	4	5	6

300

J1:

	J4			J1		
0	1	2	3	4	5	6

300 + 200

J3:

J3	J4			J1		
0	1	2	3	4	5	6

300 + 200 + 190

J2:

J3	J4	J2		J1		
0	1	2	3	4	5	6

300 + 200 + 190 + 180

J5:

J3	J4	J2	J5	J1		
0	1	2	3	4	5	6

300 + 200 + 190 + 180
+ 120

J6:

J3	J4	J2	J5	J1		
0	1	2	3	4	5	6

$$300 + 200 + 190 + 180 + 120$$

→ deadline is 1 but no free slot is available. So, job 6 cannot be completed

Answers:

a)

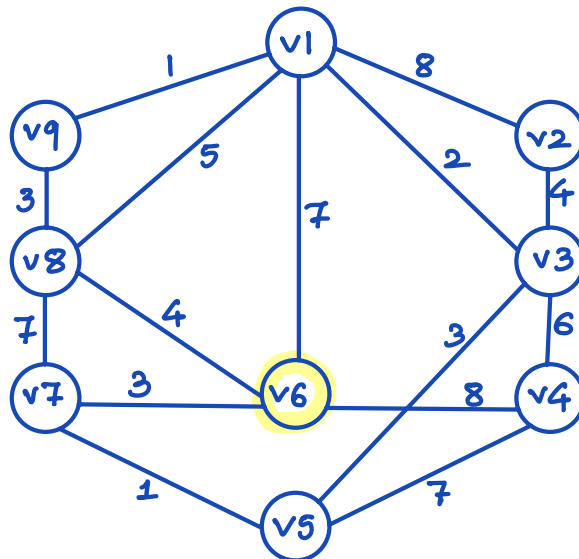
J3	J4	J2	J5	J1		
0	1	2	3	4	5	6

b) NO job 6 is not completed

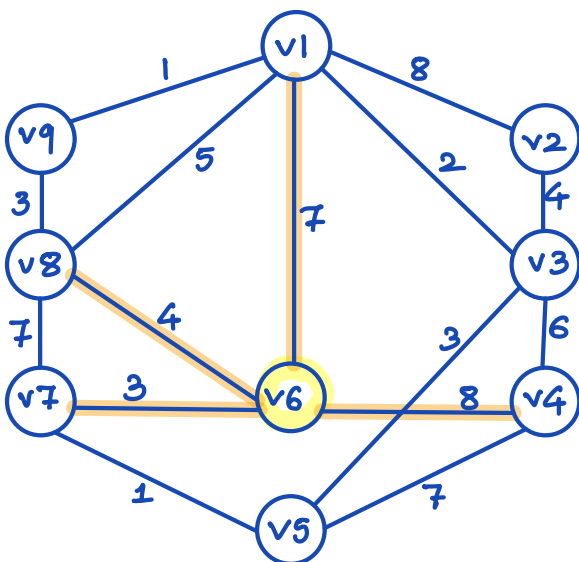
c) Profit: $300 + 200 + 190 + 180 + 120 = 990$

Ques 5:

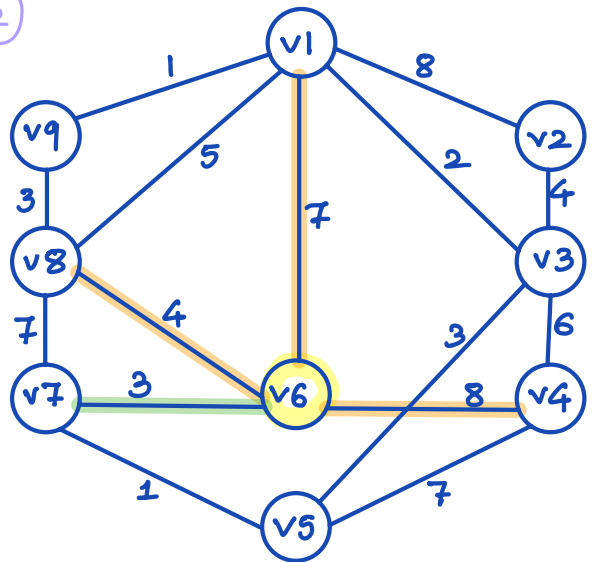
Prims:



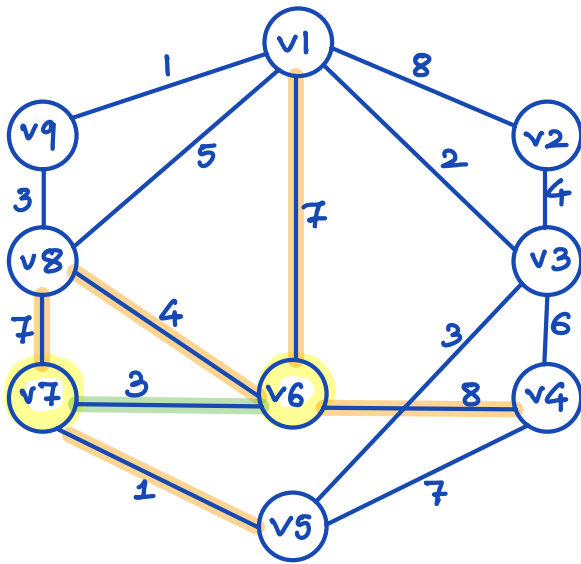
①



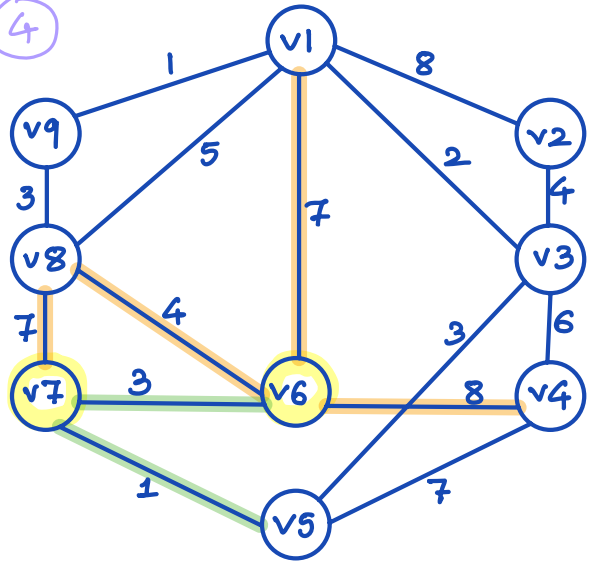
②



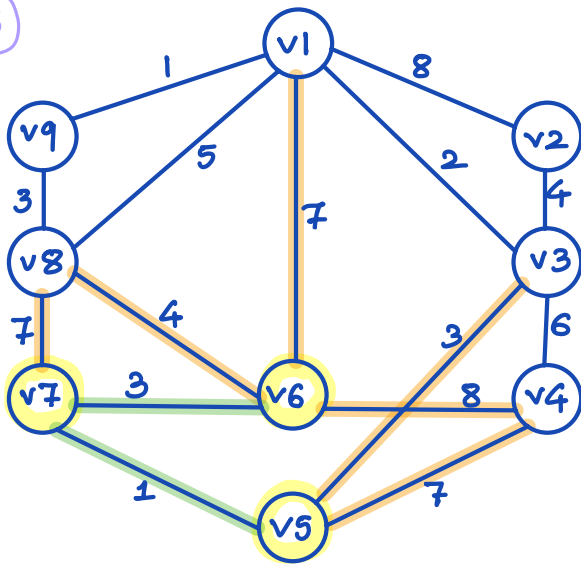
③



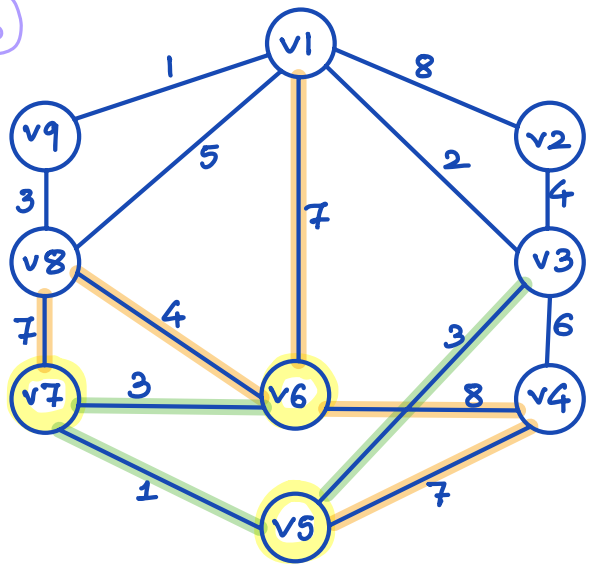
4



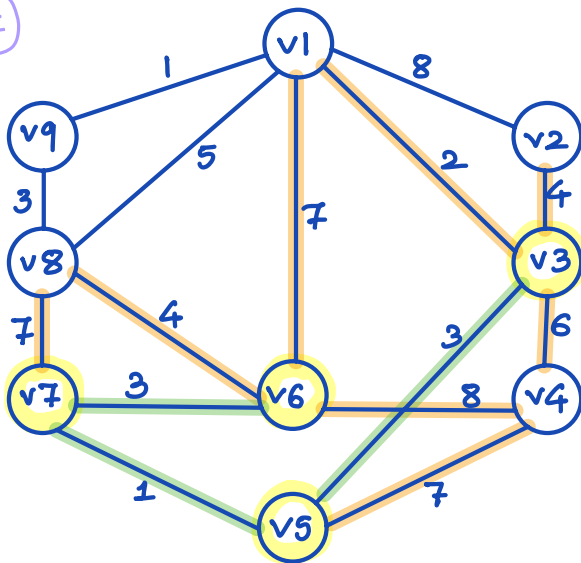
5



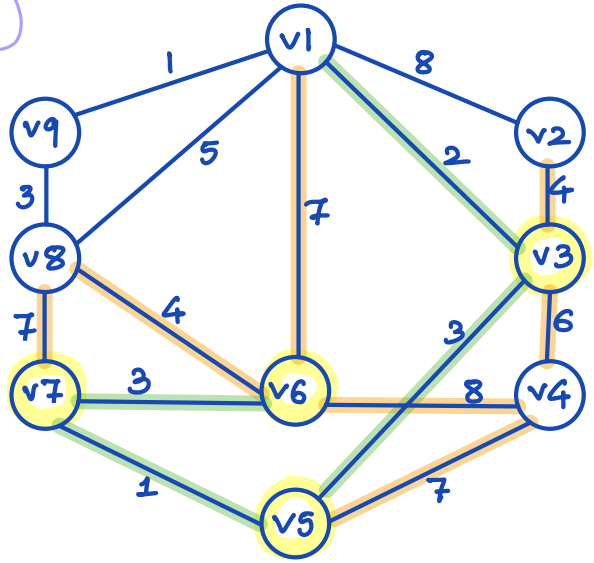
⑥



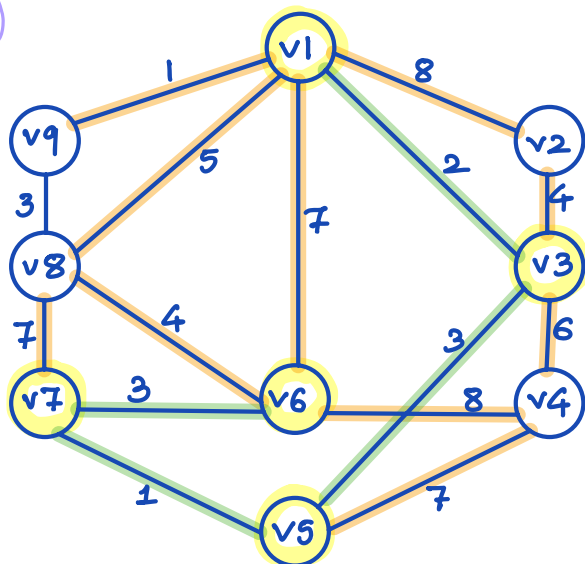
7



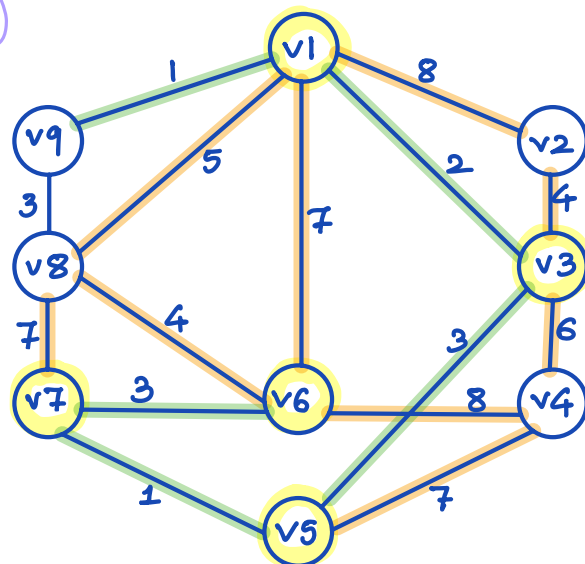
8



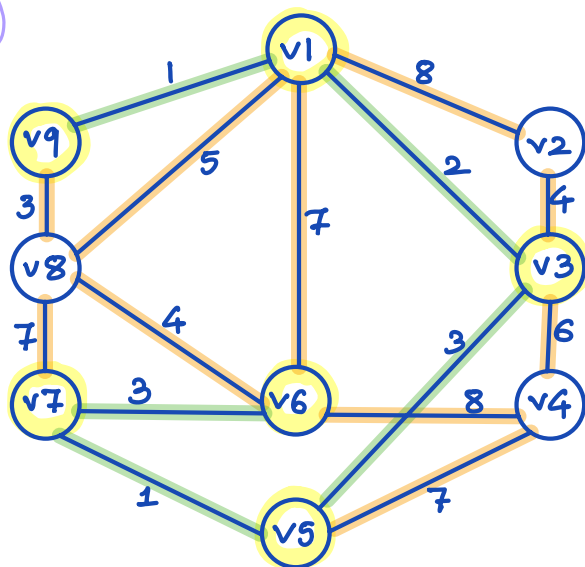
9



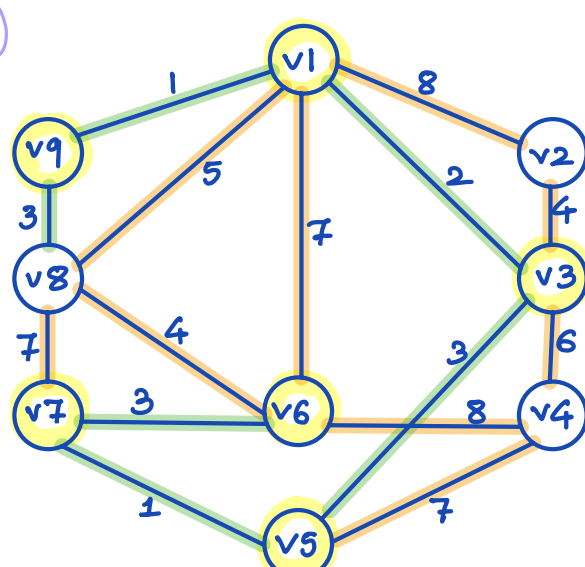
10



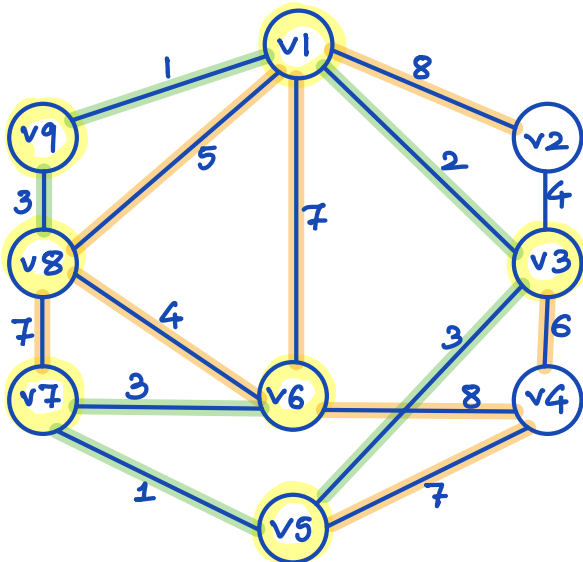
11



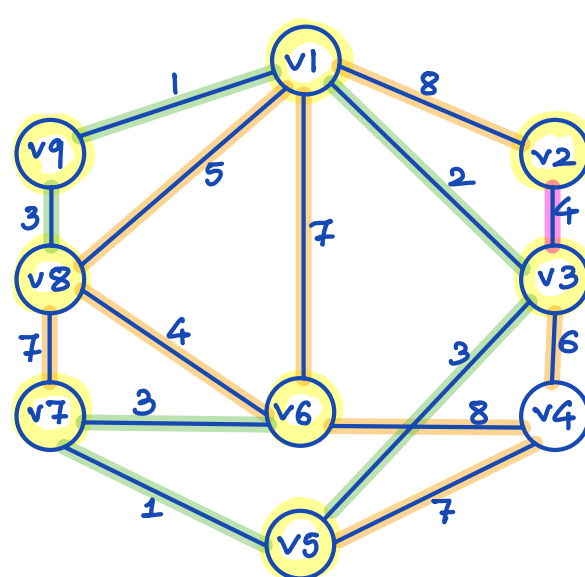
12



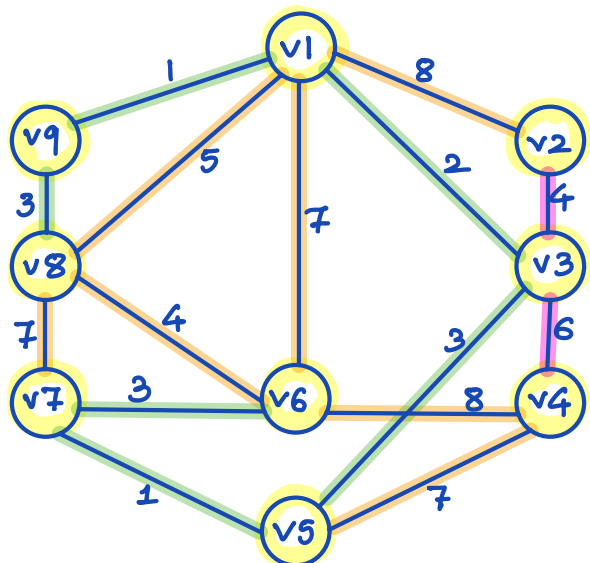
13



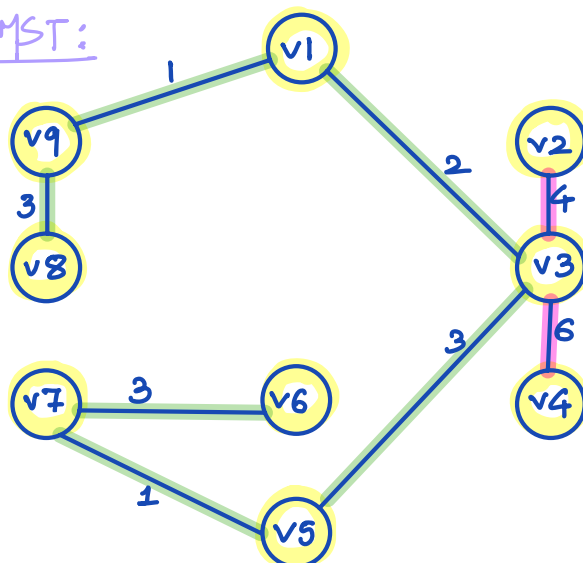
14



15



Final MST:



Total Cost : $3+1+3+2+1+3+4+6 = 23$