HW-1

CSCI-104

Q-3

Ques-3)

Part-(a)

```
void f1(int n)
{
    int i = 2;        → Θ(1)
    while(i < n){
        i = i * i;    → Θ(1)
    }
```

Since, we know that assignment operation and arithmetic operation takes constant time

So, I could say

$$T(n) = \Theta(1) + \sum_{j=1}^{\log(\log n)} \Theta(1)$$

Since, we can say $n = 16$

so, in that case we can say the time

when the loop will execute will be $i=2$, $i=4$ that means 2 times.
So, we can write 2 in terms of n as $\log(\log 16)=2$. So that is our upper bound of summation.
Thus, the final run-time is.

$$T(n) = \Theta(1) + \log(\log n)$$
$$= \log(\log n)$$

Part-b

```
void f2(int n)
{  for(int i=1; i<=n; i++){
      if( i % (int) sqrt(n) == 0)
         for( int k=0; k<pow(i,3); k++)
         {
         }
   }
}
```

In this, we can say that inner for loop will only execute when the if statement is true.

So, the initial run-time expression will look like

$$T(n) = \sum_{i=1}^{n} \left(\Theta(1) + O\left(\sum_{j=0}^{i^3} \Theta(1)\right)\right)$$

$$T(n) = \sum_{i=1}^{n}(\Theta(1)) + \sum_{i=1}^{n} O\left(\sum_{j=0}^{(i\sqrt{n})^3} \Theta(1)\right)$$

$$= \Theta(n) + \sum_{i=0}^{\sqrt{n}} \left(\Theta(i\sqrt{n})^3\right)$$

$$= \Theta(n) + \Theta\left(\frac{n(\sqrt{n}+1)^2}{4}\right) n^{3/2}$$

$$= \Theta(n) + \Theta\left(\frac{n(n+2\sqrt{n}+1)}{4}\right) n^{3/2}$$

$$= \Theta(n) + \Theta\left(\frac{n^2+2n^{3/2}+n}{4}\right) n^{3/2}$$

$= \Theta(n^{7/2})$

Thus, the final run-time is $\Theta(n^{3/2})$.

Part-C

```
for(int i=1; i <=n; i++) {
    for(int k=1; k≤n; k++) {
        if (A[k]== i) {
            for( int m=1; m≤n; m=m+m) {

            }
        }
    }
}
```

The innermost for loop will stop the iteration when $m > n$ and we know that after every iteration in the innermost loop m is doubled so we can say the number of times the loop will execute is $\log n+1$

So, the initial runtime analysis can be written as

$$T(n) = \sum_{i=1}^{n} \left(\sum_{k=1}^{n} \left(\Theta(1) + O\left(\sum_{i=1}^{(\log n)+1} \Theta(1) \right) \right) \right)$$

$$T(n) = \sum_{i=1}^{n} \left(\sum_{k=1}^{n} (\Theta(1)) + \sum_{k=1}^{n} (\Theta(\log n)) \right)$$

$$= \sum_{i=1}^{n} \left(\Theta(n) \right) + \left(\sum_{i=1}^{n} \sum_{k=1}^{n} (A[k]==i) (\Theta(\log n)) \right)$$

$$= \Theta(n^2) + \left(\sum_{k=1}^{n} \sum_{i=1}^{n} (A[k]==i) \Theta(\log n) \right)$$

$$= \Theta(n^2) + O(n\log n)$$

Thus, run-time is $\Theta(n^2)$.

Part - d

```
int f (int n)
{
    int * a = new int [10];      → Θ(1)
    int size = 10;
    for (int i = 0; i<n; i++)
    {
        if ( i==size)
        {
            int newsize = 3*size/2;
            int *b = new int [newsize]
            for(int j=0; j<size; j++) b[j]=a[j];
            delete a[];
            a=b;
            size= newsize;
        }
        a[i]=i*i ;
    }
}
```

In the above code we can say the initial run-time expression can be given as:-

The size of the function is changing in the form of $\left(\frac{3}{2}\right)^k 10 = n$. Thus, $k = \log_{\frac{3}{2}}\left(\frac{n}{10}\right)$

$$T(n) = \sum_{i=0}^{n} \Theta(1) + \sum_{j=0}^{\log_{\frac{3}{2}}\left(\frac{n}{10}\right)} \Theta\left(\left(\frac{3}{2}\right)^j 10\right)$$

$$= \Theta(n) + \Theta\left(\frac{n}{5} - 1\right)$$

$$= \Theta(n) + \Theta(n)$$

$$= \Theta(n)$$