Name: Huizhen (Olivia) Li
USC ID: 6657472930
CSCI 104 q3_Answers.

## Part (a)

Since we are not sure how many times this program will run, we set a dummy counter as k.

So, we have this

| k | i |
|---|---|
| 0 | 2 |
| 1 | 4 |
| 2 | 16 |
| 3 | 256 |
| | ⋮ |

```
void f1(int n)
{
    int i=2;
    while(i < n){
        /* do something that takes O(1) time */
        i = i*i;
    }
}
```

therefore, when it's the nth time, $i = 2^{2^k}$.

$$2^{2^k} = n$$

$$\log(2^{2^k}) = \log(n)$$

$$2^k \log(2) = \log(n)$$

$$2^k = \frac{\log(n)}{\log(2)}$$

$$k(\log 2) = \log\left(\frac{\log n}{\log 2}\right)$$

$$k = \frac{\log(\log n) - \log(\log 2)}{\log 2}$$

$$= \log(\log n)$$

In conclusion:

$$T(n) = \sum_{k=0}^{\log(\log n)-1} (\theta(1)) = \theta(\log(\log(n)))$$

## Part (b)

In the first for loop, there are n elements, after entering the if statement, there'll be $\sqrt{n}$ elements.

```
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++) {
                /* do something that takes O(1) time */
            }
        }
    }
}
```

| x | i |
|---|---|
| 1 | $\sqrt{n}$ |
| 2 | $2\sqrt{n}$ |
| 3 | $3\sqrt{n}$ |
| 4 | $4\sqrt{n}$ |
| x | $x\sqrt{n}$ |

and inside of the if statement, runtime is $\theta(i^3)$

$$x(\sqrt{n}) = n$$

$$x = \frac{n}{\sqrt{n}} = \sqrt{n}$$

$$T(n) = \sum_{i=1}^{\sqrt{n}} (i \cdot \sqrt{n})^3 = (\sqrt{n})^3 \sum_{i=1}^{\sqrt{n}} i^3$$

$$= (\sqrt{n})^3 \cdot \theta(\sqrt{n})^4$$

$$= \theta(n^{\frac{7}{2}})$$

## Part (c)

There are two undirect for loops, considering the worst case scenario, each for loop runs $n$. times therefore $\theta(n^2)$

Inside the for loop:

```
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){
        // do something that takes O(1) time
        // Assume the contents of the A[] array are not changed
      }
    }
  }
}
```

| K | m |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 4 |
| 4 | 8 |
| k | $2^k$ |

since loop stops when $k \le n$,

$$2^k = n$$
$$\log(2^k) = \log(n)$$
$$k \log(2) = \log n$$
$$k = \frac{\log(n)}{\log(2)}$$
$$\sim \log(n)$$

Therefore,

$$\text{Runtime } T(0) = \theta(n^2) + \sum_{j=1}^{2^k} \theta(1)$$
$$= \theta(n^2) + \theta(\log(n))$$
$$\sim \theta(n^2)$$

---

## Part (d)

The first for loop, runs $n$ times, without the if statement, runs $\theta(1)$ times therefore is $n \cdot \theta(1) = \theta(n)$.

Inside the if statement,

```
int f (int n)
{
  int *a = new int [10];
  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;   → θ(1)
    }
}
```

| K | i |
|---|---|
| 1 | 10 |
| 2 | 15 |
| 3 | 22 |
| 4 | 33 |
| k | $10\left(\frac{3}{2}\right)^{k-1}$ |

$$10\left(\frac{3}{2}\right)^{k-1} = n.$$
$$\left(\frac{3}{2}\right)^{k-1} = \frac{n}{10}$$
$$\log\left(\frac{3}{2}\right)^{k-1} = \log\left(\frac{n}{10}\right)$$
$$(k-1) = \frac{\log\left(\frac{n}{10}\right)}{\log\left(\frac{3}{2}\right)}$$
$$k = \frac{\log\left(\frac{n}{10}\right)}{\log\left(\frac{3}{2}\right)} + 1$$
$$\sim \log\left(\frac{n}{10}\right).$$

Again, inside the if statement, to for loop will be executed, $\log\left(\frac{n}{10}\right)$ times,

To be concluded:

$$T(n) = \theta(n) + \sum_{n=0}^{\log\left(\frac{n}{10}\right)} 10\left(\frac{3}{2}\right)^n$$
$$= \theta(n) + 10 \sum_{n=0}^{\log\left(\frac{n}{10}\right)} \left(\frac{3}{2}\right)^n$$
$$= \theta(n) + 10 \cdot \theta\left[\left(\frac{3}{2}\right)^{\log\left(\frac{n}{10}\right)} - 1\right] \cdot 2$$
$$\sim \theta(n)$$