Big Θ Runtime Analysis

Part (a)

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

for (int i = 2 ; i < n ; i = i×i)

i	K
2	0
4	1
16	2
256	3
⋮	⋮

$i = 2$

$i = 2 \times 2 = 2^2$

$2^2 \times 2^2 = 2^4$

$2^4 \cdot 2^4 = 2^8$

⋮

$$2^{2^k} \geq n$$

When is $2^{2^k} = n$?

$$\log_2(i) = \log_2 2^{2^k} = \log_2(n)$$

$$2^k = \log_2(n)$$

$$\log_2 2^k = \log(\log_2(n))$$

$$k \cdot \log_2 2 = \log(\log_2(n))$$

$$k = \log(\log_2(n))$$

$\Theta(\log_2(\log_2(n)))$ since no data dependency

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

Outer loop runs n times, inner runs $\sqrt{n}$

$$\sum_{i=1}^{n} \left(\theta(1) + \sum_{k=0}^{\sqrt{n}} i^3 \right)$$

i	i^3
1	1
2	8
3	27
$\vdots$	$\vdots$

$i = k\sqrt{n}$

$$= \sum_{i=1}^{n} \theta(1) + \sum_{i=1}^{n} \sum_{k=0}^{\sqrt{n}} (k\sqrt{n})^3$$

$$= \theta(n) + (\sqrt{n})^3 \sum_{k=0}^{\sqrt{n}} k^3$$

$$= \theta(n) + (\sqrt{n})^3 \left(\frac{n+\sqrt{n}}{2} \right)^2$$

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

$$= \theta(n) + \frac{n^{\frac{5}{2}} + 2n^{\frac{4}{2}} + n^{\frac{5}{2}}}{4} \Rightarrow \theta(n^{5/2})$$

Part (c)

```
for(int i=1; i <= n; i++){          O(n)
  for(int k=1; k <= n; k++){        O(n)
    if( A[k] == i){                 O(1)
      for(int m=1; m <= n; m=m+m){  O(n)
        // do something that takes O(1) time
        // Assume the contents of the A[] array are not changed
      }
    }
  }
}
```

$m = 2^n$

$\log_2 n = m$

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

$$= \theta(n^2) + n \sum_{m=1}^{n} O(1)$$

$$= \theta(n^2) + O(n \cdot \log_2 n)$$

$$= \theta(n^2)$$

Part d

```
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;
     }
}
```

$i = 10 \times \frac{3}{2} s$

$n = 10 \times \frac{3}{2} s$

$s = \log_{\frac{3}{2}} \frac{n}{10}$

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

$$= \sum_{i=0}^{n} \Theta(1) + \sum_{i} \sum_{j=1}^{10 \times \frac{3}{2} k} \Theta(1)$$

$$= \Theta(n) + \sum_{k=0}^{\log_{\frac{3}{2}} \frac{n}{10}} \Theta\left(10 \times \frac{3}{2} k\right)$$

$$= \Theta(n) + 10\left(\frac{3}{2}\right)^{\log_{\frac{3}{2}} \frac{n}{10}} = \Theta(n)$$