Problem 3: Runtime Analysis

Part (A)

```
void f1(int n)
{
    int i=2;  = 1
    while(i < n){ = log2n + 1
        /* do something that takes O(1) time */  = 1
        i = i*i;    = log2n
    }
}
```

$T = 1 + log_2 n + 1 + 1 + log_2 n$

$T(n) = 2(log_2 n)+3$

How I found the amount of iterations:

i=2

2*2 = 4

2^2 * 2^2 = 2^4

2^4 * 2^4 = 2^8

2^8 * 2^8 = 2^16

…

2^x

The while loop will execute 2^x times.

The while loop will terminate when i >= n

So assuming i >=n (causing termination)

then i = 2^x

therefore 2^x >= n

if 2^x = n

$x = log_2 n$

so the while loop will execute $log_2 n$ times.

$T = 2(log_2 n)+3$

$c_1 g(log_2 n) \leq 2(log_2 n) + 3 \leq c_2 g(log_2 n)$

=4.64 =7.64 =9.29

Let $c_1 = 2$ and $c_2 = 4$ and n = 5 for example… this inequality equation would evaluate to true.

ANSWER: Big Theta is $\theta(log_2 n)$

Part(B)

```
void f2(int n)
{
    for(int i=1; i <= n; i++){     = n
        if( (i % (int)sqrt(n)) == 0){
            for(int k=0; k < pow(i,3); k++) {   = √n * i^3
              /* do something that takes O(1) time */  =1
            }
        }
    }
}
```

For loop iterations:
The inner for loop will execute i^3 times.

How many times the if Statement is true:
When n = 5, inner for loop executes 2 times
When n = 10, inner for loop executes 3 times
When n = 16, inner for loop executes 4 times

Inner for loop executes $\sqrt{n}$ * i^3 times.
amount of times if statement is met is ($\sqrt{n}$) * amount of times inner for loop iterates (i^3)

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

$c_1 g(\sqrt{n})$ ≤ $\sqrt{n}$ + ≤ $c_2 g(\sqrt{n})$

=1.11 =2.23 =8.94

Let c_1 = .5 and c_2 = 4 and n = 5 for example… this inequality equation would evaluate to true.

ANSWER: Big Theta is $\theta(\sqrt{n})$

Part (C)

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

T = n * n * $\sqrt{n}$ + 1

T = n^2 $\sqrt{n}$ + 1

T(n) = n^2

$c_1 g(n^2)$ ≤ n^2 + ≤ $c_2 g(n^2)$

=12.5 = 25 =50

Let c_1 = .5 and c_2 = 2 and n = 5 for example… this inequality equation would evaluate to true.

ANSWER: Big Theta is $\theta(n^2)$

Part (D)

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

T(n) = 1 + 1 + n(1+1+1+1+1+1+1)
T(n) = 2 + 7n
     = n
```

ANSWER: Big Theta is $\theta(n)$