3a)

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

$\underline{n = 32}$

1. $i = 2 \quad n = 32$
2. $i = 4 \quad n = 32$
3. $i = 16 \quad n = 32$
4. $i = 32 \quad n = 32$

The loop starts at 2 and squares itself/doubles everytime the loop iterates. If the initial value of $n = 32$, then it takes 4 iterations for i to reach 32

$\therefore \boxed{\Theta(\log i)}$

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 */
            }
        }
    }
}
```

$\theta(n)$

$\theta(1)$

$\theta(i^3)$

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

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

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

$1 + n + 1 + n = 2n+2$

θ

$\theta(1)$

$\theta(\log n)$

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

$$T(n) = \theta(n^2) + \sum_{k=1}^{n} (\log n)$$

$$= \theta(n^2) + \theta(n \log n) = \boxed{\theta(n^2)}$$

d'

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

$\theta(size) = \theta(3^{k/2}) \rightarrow newsize = 3size/2$

$T(n) = \sum_{k=0}^{\log_3 n} [\theta(3^{k/2}) + (n + \log_3 n)]$

$= \theta(3^{\log_3 n/2}) = \theta(n/2)$

$= \theta(n)$

The first loop runs $n+1$ times. Assuming it loop always executes, the second loop would run 10 times. Size updates $(3^{size/2})$ which is $\theta(3^{\log_3 n})$