a)

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

k	1	2	3	n = i^2^k, solve for k k = $\log_2(\log_i(n))$
i	i^2	i^4	i^8	i^2^k = n

$$\sum_{k=0}^{\log_2(\log_i(n))} \theta(1) = \theta(\log(\log(n))$$

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 */
            }
        }
    }
}
```

j	1	2	3	j	$\sqrt{n}$	j goes from 1 to $\sqrt{n}$
i	$(\sqrt{n})^3$	$(2\sqrt{n})^3$	$(3\sqrt{n})^3$	$(j\sqrt{n})^3$	$(\sqrt{n}\sqrt{n})^3$	

$$\sum_{i=2}^{n} (\theta(1)) + O(\sum_{k=0}^{i^3} \theta(1))$$

$$= \sum_{i=2}^{n} (\theta(1)) + \sum_{j=1}^{\sqrt{n}} (\sum_{k=0}^{i\wedge 3} \theta(1))$$

$$= \theta(n) + \sum_{j=1}^{\sqrt{n}} \theta((j*\sqrt{n})^3)$$

$$= \theta(n) + \sum_{j=0}^{n} \theta(j^3*n^{3/2})$$

$$= \theta(n) + \theta(n^{3/2})*\sum_{j=1}^{\sqrt{n}} \theta(j^3)$$

$$= \theta(n) + \theta(n^{3/2}) * \theta((n^{1/2})^4)$$

$$= \theta(n) + \theta(n^{4/2 + 3/2})$$

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

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

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
      }
    }
  }
}
```

p	1	2	3	p	$n = 2^{p-1}*m$ $2^p = n/2m$ $p = \log_2(n/2m)$
m	m	2*m	4*m	$2^{p-1}*m$	$2^{p-1}*m = n$

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

$$= \sum_{i=1}^{n} ((\sum_{k=1}^{n} (\theta(1) + (\sum_{p=1}^{\log_2(n/2m)}\theta(1)))$$

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

$$= \sum_{i=1}^{n} (\theta(n) + \theta(\log n))$$

$$= \sum_{i=1}^{n} (\theta(n)) + \sum_{i=1}^{n}(\theta(\log n))$$

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

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;
    }
}
```

k	1	2	3	k	size = 10 $10*1.5^{k-1} = n$ $10*1.5^{k}/1.5 = n$ $1.5^{k} = 0.15*n$ $k = \log_{1.5}(0.15n)$
size	size	size * 1.5	(size*1.5)*1.5	size * 1.5^{k-1}	n

$$\sum_{i=1}^{n}\theta(1) + O\left(\sum_{j=0}^{j=size}\theta(1)\right)$$

$$= \sum_{i=1}^{n}\theta(1) + \left(\sum_{k=1}^{\log_{1.5}(0.15n)}\theta(size * 1.5^{k-1})\right)$$

$$= \sum_{i=1}^{n}\theta(1) + \left(\sum_{k=1}^{\log_{1.5}(0.15n)}\theta(1.5^{k})\right)$$

$$= \theta(n) + \theta(1.5^{\log1.5(0.15n)})$$

$$= \theta(n) + \theta(n) = \theta(n)$$