

a.

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

$$2^i \leq n$$

$$i \leq \log_2 n$$

$$\sum_{i=1}^{\log_2 n} \theta(1)$$

$$\boxed{\Theta(\log n)}$$

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 */
            }
        }
    }
}
```

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

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

$$n = 9$$

$$= \sum_{i=1}^9 \theta(1) + \sum_{i=1}^9 \sum_{k=0}^{i^3} \theta(1) + \sum_{i=1}^9 \theta(1) + \sum_{i=1}^9 \theta(1)$$

$$= \theta(n) + n(\theta(1) + \theta(1) + \theta(1) + \theta(1) + \theta(1) + \theta(1) + \theta(1) + \theta(1) + \theta(1))$$

$$= \theta(n) + n(8\theta(1))$$

$$= \theta(n) + 8n\theta(1)$$

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

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
            }
        }
    }
}
```

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

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

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

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

$$= \sum_{i=1}^n \theta(n^2)$$

$$= \theta(n^3)$$

(assume it runs every time)

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

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

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

$$= \sum_{i=1}^n \theta(n^2)$$

$$= \theta(n^3)$$

$$O + \Omega \text{ both } (n^3)$$

$$\therefore \boxed{\Theta(n^3)}$$

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;
    }
}
```

$\theta(1)$
 4, size size
 is constant
 ...
 even if it grows
 exponentially

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

$$\theta(1) + \sum_{i=1}^n (\theta(1) + 10\theta(1))$$

$$\theta(1) + \theta(n)$$

$$\boxed{\theta(n)}$$