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

runs n times

$i = i \times i$

$i^2 = 2^2$

$i^2 = 2^{2^k}$ $\quad k < \log(\log n)$

$2^{2^k} < n$

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

```
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 */
            }
        }
    }
}
```

runs n times

runs i^3 times

$$= \sum_{i=1}^{n} \Theta(1) + \sum_{i=1}^{\sqrt{n}} \Theta(i^3)$$

$$= \Theta(n) + \sum_{K=1}^{\sqrt{n}} \Theta\left(K \cdot (\sqrt{n})^3\right)$$

$$= \Theta(n) + (\sqrt{n})^3 \sum_{K=1}^{\sqrt{n}} K^3$$

$$= \Theta(n) + (\sqrt{n})^3 \Theta(\sqrt{n})^4$$

$$= \Theta(n) + \Theta(\sqrt{n})^4$$

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

```
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
      }
    }
  }
}
```

runs n times

runs n times

m = 2m

runs log n times

$m = 1, 2, 4, 8 \ldots 2^K$

$2^K = n$

$K = \log n$

Assuming A satisfies the case when i=1 and K loops through

$A[K] = i \Rightarrow A[1] = 1$

$A[2] = 1$

$\vdots$

$A[K] = 1$

The inner for loop is executed $K = n$ times since the contents of the array is fixed.

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

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

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

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

```
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	10
40	10 + 40
160	10 + 40 + 160
640	10 + 40 + 160 + 640
2560	10 + 40 + 160 + 640 + 2560
⋮	⋮
i	10 + 40 + 160 + 640 + 2560 + i

K	i
0	10
1	40
2	160
3	640
4	2560

$i = 10 \times 4^K$

$n = 10 \times 4^K$

$K = \log_4 n/10$

$$\sum_{i=0}^{n} \theta(2) + \sum_{K=0}^{\log_4 n/10} \left(\theta(1) + \sum_{j=1}^{10\times 4^K} \theta(1)\right)$$

$$= \theta(2n) + \sum_{K=0}^{\log_4 n/10} \theta(1) + \sum_{K=0}^{\log_4 n/10} \theta(10 \times 4^K)$$

$$= \theta(2n) + \theta(\log_4 n/10) + 10 \sum_{K=0}^{\log_4 n/10} \theta(4^K)$$

$$= \theta(2n) + \theta(\log_4 n/10) + 10\,\theta(n/10)$$

$$= \theta(2n + \log_4(n/10) + n)$$

$$= \theta(3n) = \boxed{\theta(n)}$$