Problem 3

```
void F1(int n)
{
  int i=2;
  while (i<n) {
     // do something that takes O(1) //
     i= i*i
  }
}
```

Set n to be 65,536.

	i = i*i
i=2	2·2 = 4
i=4	4·4 = 16
i=16	16·16 = 256
i=256	256·256 ≈ 65k

2×2 = 4
4× 4 = 16
16 × 16 = 256

① 2×2 = 4
② 4×2 = 8
③ 8×2 = 16
④ 16×2 = 32

⑤ 32 × 2 = 64
⑥ 64×2 = 128
⑦ 128×2 = 256

$$\sum_{i=2}^{\log(n)} \log(n) = \underline{\log(\log n)}$$

```
b)  void f2(int n)
    {
     for (int i= 1; i<=n; i++){        O(n)
        if(i % (int)sqrt(n)) ==0) {
            for(int k=0; k< pow(i,3); k++){
                // O(1) time//                        O(1)
            }
        }
     }
    }
```

$i = \sqrt{n}, 2\sqrt{n}, 3\sqrt{n} \dots n \quad \sqrt{n}\cdot\sqrt{n}$

$\sum_{i} \sum_{k=0}^{i^3} O(1) \quad i^3 \quad \sum_{i} i^3 = \sum_{j=1}^{\sqrt{n}} (j\sqrt{n})^3$

$\sum_{j=1}^{\sqrt{n}} (i\sqrt{n})^3 \Rightarrow (i^3\sqrt{n}^3)(\sqrt{n})$

$i^3\sqrt{n}\;\sqrt{n}^4$

$i^3 n^{5/2}\cdot n = \underline{\Theta(n^{7/2})}$

```
c) for(int i=1, i≤n ;i++){               O(n)
     for(int k=1; k≤n; k++){  ——————→
       if (A[k] == i) {
O(log n)  for (int m=1; m≤n; m+m) {   m: 1, 2, 4, 8, .... 2^m
             //O(1) time //
          }
       }
     }
   }
```

$\sum_{k=1}^{n} O(n) = n^2$

$n\log n + n^2$ If everything runs

or $\underline{\boxed{n^2}}$ if

```
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 ;
      }
   }
```

size=10

$\Theta(n)$ if $i \neq size$

$\left(\frac{3}{2}\right)^n$

$\left(\frac{3}{2}\right)10, \left(\frac{3}{2}\right)^2 10, \left(\frac{3}{2}\right)^3 10 \ldots$