Part a)

```
void f1(int n)
{
    int i=2
    while (i<n){
        i=i×i      O(1)
    }
}
```

x	0	1	2	3
i	2 →	4 →	16 →	256

growth of function 2^{2^x}

$2^{2^{(0)}} = 2^1 = 2$

$2^{2^{(1)}} = 2^2 = 4$

$n = 2^{2^x}$

$\log n = 2^x$

$\log(\log n) = x$

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

code runs $\log(\log(n))$, so O(1) code

$1 \cdot \log(\log(n)) = \log(\log(n))$

part b)

```
void f2(int n)
{
    for (int i=1; i<=n; i++){          n times
        if ( i%(int)sqrt(n) ==0) {
            for(int k=0  k<pow(i,3); k++){     i^3
                Θ(1)
            }
        }
    }
}
```

(As you can see, I did this after the 9/1 lecture)

if statement is true $\sqrt{n}$ times

ex	$n=9$	$\sqrt{9}=3$	i : 1-9	3, 6, 9
	$n=16$	$\sqrt{16}=4$	i : 1-16	4, 8, 12, 16

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

j	1	2	3	j	j = ?
i	$\sqrt{n}$	$2\sqrt{n}$	$3\sqrt{n}$	$j\sqrt{n}$	$i=n$

$i = j\sqrt{n}$

$j\sqrt{n} = n$

$j = \sqrt{n}$

turns into $\sum_{j=1}^{\sqrt{n}}$

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

$$= n + \sum_{j=1}^{\sqrt{n}} i^3$$

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

$$n + \sum_{j=1}^{\sqrt{n}} n^{3/2} j^3$$

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

$$\Theta(n) + n^{3/2}\,\Theta(\sqrt{n}^{\,4})$$

$$\Theta(n) + \Theta(n^{3/2} \cdot n^2)$$

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

Part c)

runs n times

```
for (int i=1, i<=n , i++) {          runs n times
    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
            }
        }
    }
}
```

how many times do have to add m together to get n

worse case scenario all elements match i so code runs n times

x = amount of iterations

x	1	2	3	4	5	6
m	1	2	4	8	16	32

$n = 2^{x-1}$

$\log n = x - 1$

$\log n + 1 = x \rightarrow \log n$

$m = 2^{i-1}$

$2^{1-1} = 2^0 = 1$

runtime $= n \cdot n \cdot \log n$ $\boxed{\Theta(n^2 \log n)}$

arithmetic series

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

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

geometric series

$$\sum_{i=0}^{n} c^i = \frac{c^{n+1} - 1}{c - 1} = \Theta(c^n)$$

$$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$$

because

$$\sum_{i=0}^{\lg(n)} \sum_{j=0}^{2^i - 1} \Theta(1)$$

$$\sum_{i=0}^{\log(n)} \Theta(2^i) = \frac{2^{\log(n)+1} - 1}{2 - 1} =$$

$$= \frac{2^{\log_2(n)}(2)^1 - 1}{1}$$

$= 2n - 1$

$= n$

Part 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 = 3/2 (size);
            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;
    }
}
```

O(1) (first two lines)

runs n times (for loop)

if (i == size) — executes $\log_{\frac{3}{2}}(n)$ times

O(1)

runs size times (inner for loop)

O(1)

a[i] = i*i; — O(1)

would be fractions, but integer math floors it

1. size = 10, 15, 22, 33

size	10	15	22	33
times it resizes	1	2	3	4

2. $15 = 10\left(\frac{3}{2}\right)$

$22 \approx 10\left(\frac{3}{2}\right)^2$

$33 \approx 10\left(\frac{3}{2}\right)^3$

3. $10 \cdot \left(\frac{3}{2}\right)^x = n$

$\left(\frac{3}{2}\right)^x = \frac{n}{10}$

$x = \log_{\frac{3}{2}}\left(\frac{n}{10}\right)$

$= \log_{\frac{3}{2}}(n) - \log_{\frac{3}{2}}(10)$

$= \log_{\frac{3}{2}}(n)$

4. $$\sum_{i=0}^{\log_{3/2}(n)} \sum_{j=0}^{10\left(\frac{3}{2}\right)^{A}} 1$$

$$\sum_{i=0}^{\log_{3/2}(n)} 10\left(\frac{3}{2}\right)^n = 10\sum_{i=0}^{\log_{3/2}(n)} \left(\frac{3}{2}\right)^n$$

$$= 10\left(\frac{\frac{3}{2}^{\log_{\frac{3}{2}}(n)+1} - 1}{\frac{3}{2} - 1}\right) \quad \text{(geometric series)}$$

$$= 10\left(\frac{\frac{3}{2}^{\log_{\frac{3}{2}}(n)}\left(\frac{3}{2}\right) - 1}{\frac{1}{2}}\right)$$

$$= 10\left(\frac{\frac{3}{2}n - 1}{\frac{1}{2}}\right) = 20\left(\frac{3}{2}n - 1\right) = 30n - 1 = \boxed{\Theta(n)}$$