Part A:

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

K	I		
0	2	2^1	2^{2^0}
1	4	2^2	2^{2^1}
2	16	2^4	2^{2^2}
3	256	2^8	2^{2^3}

Relation is 2^{2^k}

$2^{2^k} < n$

$2^k < log_2 n$

$k < log_2(log_2(n))$

$\Theta(1) + \Theta log(log(n))$

Answer: $\Theta log(log(n))$

Part 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 */
```

```
                        }
                }
        }
}
```

K	I	
16	1	1^3 times loop is executed
	4	4^3 times loop is executed
	8	8^3 times loop is executed
	12	12^3 times loop is executed
	16	16^3 times loop is executed

When K = 16, total number of times the inner loop is executed =
$4^3 + 8^3 + 12^3 + 16^3$

$= (1 \times \sqrt{16}^3) + (2 \times \sqrt{16}^3 + (3 \times \sqrt{16}^3) + (4 \times \sqrt{16}^3)$

$= \Sigma_{k=1}^{\sqrt{n}} \Theta(k\sqrt{n})^3$

$= \Sigma_{k=1}^{\sqrt{n}} \Theta(k^3 \sqrt{n}^3)$

= $\sqrt{n}^3 \Sigma_{k=1}^{\sqrt{n}} \Theta(k^3)$

$= \sqrt{n}^3 \Theta(\sqrt{n}^4) = \Theta(n^{7/2})$

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

Part 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
                            }
                    }
          }
}
```

Z	M	
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
5	32	2^5

$\Sigma_{i=1}^{n}\Sigma_{k=1}^{n}(O(1) + \Sigma_{m=1,2,3...}^{n}O(1))$

$\Sigma_{i=1}^{n}\Sigma_{k=1}^{n}O(1) + n\Sigma_{m=1,2,3...}^{n}O(1)$

$$m < n$$

$$2^z = m$$

$$2^z < n$$

$$z \times log(2) < log(n)$$

$$z < log(n)$$

$\Sigma_{i=1}^{n}\Sigma_{k=1}^{n}O(1) + n\Sigma_{m=1,2,3...}^{n}log(n)$

Answer: $\Theta(n^2) + O(n \times log(n))$

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*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;
        }
}
```

n	i	size	times executed
30	0	10	
	10	15	10
	15	22	15
	22	33	22

Total # of times for loop executed when n = 30: $10 + 15 + 22$

$$= 10(\frac{3}{2}^0) + 10(\frac{3}{2}^1) + 10(\frac{3}{2}^2)$$

$$= \Sigma_{k=10}^{?}(10 \times (\frac{3}{2}^k))$$

$$10(\frac{3}{2}^k) < n$$

$$(\frac{3}{2}^k) < \frac{n}{10}$$

$$k \times log_{\frac{3}{2}}(\frac{3}{2}) < log_{\frac{3}{2}}(\frac{n}{10})$$

$$k < log_{\frac{3}{2}}(\frac{n}{10})$$

$$\Sigma_{k=0}^{log_{\frac{3}{2}}(\frac{n}{10})}(\Theta(10 \times \frac{\mathbf{3}}{\mathbf{2}}^{k}))$$

$$= \Theta(\frac{\mathbf{3}}{\mathbf{2}})^{log_{\frac{3}{2}}\frac{n}{10}}$$

$$= \Theta(\frac{n}{10}) = \Theta(n)$$

Answer: $\Theta(n)$