PROBLEM 3: RUNTIME ANALYSIS

a)
```
int i=a;
while(i<n){
    i = i*i;
}
```

$a = a^* a =$
$4 = 4^* 4 = 16$

a^{2^k}

k	j
0	2
1	4
2	16

$a^{2^k} \geq n$

$\log(a^{2^k}) \geq \log(n)$

$2^k \cancel{\log(a)}^{1} \geq \log(n)$

$\log(2^k) \geq \log(\log(n))$

$k \cancel{\log(2)}^{1} \geq \log(\log(n))$

$k \geq \log(\log(n))$

$\Theta(\log(\log(n)))$

b)
```
For(int i=1; i<=n; i++){
    if((i%(int)sqrt(n))==0){
        For(int k=0; k<pow(i,3); k++){
            O(1)
        }
    }
}
```

n	$\sqrt{n}$	"if" triggers
9	3	3, 6, 9
16	4	4, 8, 12, 16

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

j	1	2	3 ...
i	$1\sqrt{n}$	$2\sqrt{n}$	$3\sqrt{n}$ ...

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

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

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

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

$$= \Theta(n) + \Theta(n^2) = \Theta(n^2)$$

j	j
i	$j\sqrt{n}$

→ also stops when $i = n$

$j\sqrt{n} = n$

$j = \frac{n}{\sqrt{n}} = \frac{n\sqrt{n}}{n} = \sqrt{n}$

c) `for (int i=1; i <= n; i++) {` → n times

`for (int k=1; k <= n; k++) {` → n times

`if (A[k] == 1) {` → worst case: i times, best case: 0 times

`for (int m=1; m <= n; m = m+m) {`

`}`

`}`

`}`

n = 10	n = 16	n = 19
m = 1	m = 1	m = 1
m = 2	m = 2	m = 2
m = 4	m = 4	m = 4
m = 8	m = 8	m = 8
	m = 16	m = 16

m	m + m
1	2
2	4
4	8
8	16

$$\sum_{i=1}^{n}\left(\Theta(1)+\sum_{k=1}^{n}\left(\Theta(1)+0\left(\sum_{m=1}^{\log(n)+1}\Theta(1)\right)\right)\right)$$

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

$$\sum_{i=1}^{n}\Theta(n) \qquad \sum_{i=1}^{n}\sum_{k=1}^{n}\Theta(\log(n)+1)$$

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

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

$$=\Theta(n^2\log(n))$$

$$=\Theta(n\cdot(n\log(n)+n))$$

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

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;
  }
}
```

← happens once → run time depends on size, whatever that is in each specific iteration

$\Theta(n)$