3.

(a)
```
int i=2
while(i<n){
    i=i*i
}
```

$i = 2, 4, 16, 256$ (×2, ×4, ×16, ×2, ×4, ×2)

$i = 2^{2^x}$

$2^{2^x} < n$

$$\log(2^{2^x}) < \log(n) = 2^x \log(2) < \log(n)$$

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

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

(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)
        }
    }
}
```

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

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

$$= \Theta(n) + \sum_{i=1}^{\sqrt{n}} \Theta(i^2) = \Theta(n) + \Theta(\sqrt{n}^4)$$

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

c)
```
for(int i=1; i<=n; i++){          n times
    for(int k=1; k<=n; k++){      n times
        if(A[k] == i){            worst: i times  best: 0
            for(int m=1; m<=n; m=m+m){
            }
        }
    }
}
```

$m = 1, 2, 4, 8, 16, 32, 64$ (+1, +2, +4, +8, +16, +32)

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

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

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

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

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

d)

```
int *a = new int [10];
int size = 10;
for (int i=0; i<n; i++) {        O(n)
    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;
   }
}
```

depends on size

$O(n)$