Runtime Analysis

1) a.

k		
1	2	2^1
2	4	2^2
3	16	2^4
4	256	2^8

$2^{2^k} < n$

$k < \log(\log n)$

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

b.

```
for(int i=1; i<=n; i++) { // Θ(n)
  if((i%(int)sqrt(n))==0) { // this is checked √n times
    for(int k=0; k<pow(i,3); k++) { // Θ(n^3)
      do something that takes O(1) time
```

outer loop: $\sum_{i=1}^{n}$ ($\Theta(1) + \sum_{j=1}^{\sqrt{n}} \sum_{k=0}^{i^3-1} O(1)$) simplifies to $\Theta(n) + \sum_{i=1}^{\sqrt{n}} (\Theta(i^3))$

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

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

if statement doesn't trigger

number of times if statement is true

number of times the inner for loop runs.

if n	$\sqrt{n}$	if statement is true
9	3	3, 6, 9
16	4	4, 8, 12, 16

c.

```
for(int i=1; i<=n; i++) { // Σ_{i=1}^{n} O(1) = Θ(n)
  for(int k=1; k<=n; k++) { // Σ_{k=1}^{n} O(1) = Θ(n)
    if(A[k]==i) // worst case scenario, if statement is true Θ(n) times.
for(int m=1; m<=n; m=m+m) {
```

k	m
1	1
2	2
3	4
4	8
5	16

$2^{k-1} \le n$

$k-1 \le \log(n)$

$k \le \log(n)+1$

$\Theta(\log(n))$

first 2 for loops: $T(n) = \Theta(n^2)$ × (worst case scenario × runtime of innermost for loop) $+ \Theta(n \cdot \log(n))$

$\boxed{T(n) = \Theta(n^2)}$

d.

k	
0	10
1	15
2	22
3	33

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

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

$k \log(\frac{3}{2}) = \log(n/10)$

$k = \log_{3/2}(n/10)$

first for loop: $\Theta(n) + \Theta(10 \sum_{j=0}^{k} (\frac{3}{2})^j)$

$\Theta(n) + \Theta(10 \sum_{j=0}^{\log_{3/2}(\frac{n}{10})} (\frac{3}{2})^j)$

$10(\Theta(n/10)) = \Theta(n)$

$\boxed{T(n) = \Theta(n)}$