Part (a)

$i=2$, k represents the number of iterations

k	1	2	3	4	5	6	...	k
i	2^2	2^4	2^8	2^{16}	2^{32}	2^{64}	...	2^{2^k}

After the k-th iteration, $i = 2^{2^k}$

When $i = n$, we have

$$2^{2^k} = n$$
$$2^k \log 2 = \log n$$
$$k \log 2 = \log(\log n)$$
$$k = \log(\log n)$$

Thus,

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

Part(b)

when $n=1$, $\sqrt{n}=1$, $i=1 \Rightarrow$ if-statement is true

when $n=2$, $(int)\sqrt{n}=1$, $i=1 \Rightarrow$ if-statement is true

when $n=3$, $(int)\sqrt{n}=1$, $i=1 \Rightarrow$ if-statement is true

when $n=4$, $\sqrt{n}=2$, $i=2,4 \Rightarrow$ if-statement is true

$\vdots$

when $n=9$, $\sqrt{n}=3$, $i=3,6,9 \Rightarrow$ if-statement is true

$\vdots$

when $n=16$, $\sqrt{n}=4$, $i=4,8,12,16 \Rightarrow$ if-statement is true.

Thus, when $n=m$, there ~~will~~ are $\sqrt{m}$ number of i will make the if-statement to be true

$$
\begin{aligned}
T(n) &= \sum_{i=1}^{n} \Theta(1) + \sum_{i=1}^{\sqrt{n}} \sum_{k=0}^{i^3} \Theta(1) \\
&= \Theta(n) + \sum_{i=1}^{\sqrt{n}} \Theta(i^3) \\
&= \Theta(n) + \Theta(\sqrt{n}^{3+1}) \\
&= \Theta(n) + \Theta(n^2) \\
&= \Theta(n^2)
\end{aligned}
$$

Part (c)

Firstly, we need to estimate the number of times that if (A[k] == i) will be true

when $i=1$, the if-statement is checking
$A[1] == 1$?
$A[2] == 1$?
$\vdots$
$A[k] == 1$?

when $i=2$, the if-statement is checking
$A[1] == 2$?
$A[2] == 2$?
$\Downarrow$
$A[k] == 2$?

The worst case could be for every i, we can find an index k that has $A[k] == i$

Thus,

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

$m = m+m$ after each iteration

Let t represents the number of iterations

t	1	2	3	...	t
$m=1$	2	4	8	...	2^t

~~Since~~ When $m=n$,

$$2^t = n$$
$$t \log 2 = \log n$$
$$t = \log n$$

Therefore,

$$T(n) = \sum_{i=1}^{n} \left(\sum_{k=1}^{n} \Theta(1) \right) + \sum_{i=1}^{n} \left(\sum_{m=1}^{\log n} O(1) \right)$$
$$= \sum_{i=1}^{n} \left(\Theta(n) \right) + \sum_{i=1}^{n} \left(\Theta(\log n) \right)$$
$$= \Theta(n^2) + \Theta(n \log n)$$
$$= \Theta(n^2)$$

Part (d)

Firstly, analysizing the following codes ↓

```
if ( i == size)
{
    int newsize = 3* size/2;                    Θ(1)
    int* b = new int[newsize];                  Θ(1)
    for ( int j=0 ; j< size ; j++) b[j] = a[j]
    delete [] a;                                Θ(1)
    a = b                                       Θ(1)
    size = newsize;                             Θ(1)
}
```

k represent the number of iterations when if(i==size) is true

k	1	2	3	4	...	k
$i=10$	$10\times\frac{3}{2}$	$10\times(\frac{3}{2})^2$	$10\times(\frac{3}{2})^3$	$10\times(\frac{3}{2})^2$	...	$10\times(\frac{3}{2})^k$

when $i=n$, we have

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

$$\log n \approx k\log(\tfrac{3}{2})$$

$$\approx k$$

Thus,

$$T(n) = \sum_{i=1}^{n}\Theta(1) + \sum_{i=1}^{\log n}\left(\Theta(5) + \sum_{j=0}^{10\times(\frac{3}{2})^{i-1}}\Theta(1)\right) + \sum_{i=1}^{n}\Theta(1)$$

$$= \Theta(n) + \Theta(5\log n) + \sum_{i=1}^{\log n}\Theta(\tfrac{3}{2}^{i}) + \Theta(n)$$

$$= \Theta(2n) + \Theta(5\log n) + \Theta(\tfrac{3}{2}^{\log n})$$

$$= \cancel{\Theta(\tfrac{3}{2}^{\log n})}\ \Theta(n)$$