2, 4, 16, 56 $\log(\log n)$

a) $i = 2 \rightarrow O(1)$

$n = 10$ 2, 4 $\rightarrow$ 2 times $n = 5$ 2, 4 $\rightarrow$ 2 times

$n = 20$ 2, 4, 16 $\rightarrow$ 3 times $\log 1 = 0$

$n = 30$ 2, 4, 16 $\rightarrow$ 3 times $\log 10 = 1$

while $(i < n)$ $\log 100 = 2$

            $\log n$
$O(1)$ $\boxed{= \Theta(\log(\log n))}$

$i = i \times i;$

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

$\displaystyle = \sum_{i=1}^{n} \Theta(1) + \sum_{i} \sum_{k=0}^{i^3-1} \Theta(1)$ $\nearrow$ 3 times
                                                       $n = 9 \rightarrow i \% 3 == 0$

$= \Theta(n) + \sum_{i} \Theta(i^3 - 1)$ $\rightarrow$ 3, 6, 9

$= \Theta(n) + \displaystyle\sum_{k=1}^{\sqrt{n}} \Theta(i^3)$ $n = 16 \rightarrow i \% 4 == 0$

                                                       4, 8, 12, 16 $\rightarrow$ times

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

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

c)   for $(int\ i = 1;\ i <= n;\ i++) \rightarrow O(n)$

for $(int\ k = 1;\ k <= n;\ k++) \rightarrow O(n)$

if $(A[k] == i) \longrightarrow$ the   worst   case   of   this   is   only if

the   condition   is   true   in n iterations $\rightarrow O(1)$
                                                                    k is
                                                                    limited
for $(int\ m = 1;\ m <= n;\ m = m+m) \rightarrow \Theta(\log(n))$ by n

time   complexity $= O(n) \times O(n) = O(n^2)$; $O(n^2) + \Theta(\log n)$

$\boxed{= O(n^2)}$

$\displaystyle\sum_{m=1}^{n} \frac{1}{m} = \Theta\log(n)$

d)   int *a = new int [10]; → O(1)

     int size = 10;   → O(1)

      for (int i=0; i<n; i++) → O(n)

        if (i == size) → O(1)

           int newsize = 3*size/2 → O(1)   loop is running
                                                     size times
           int *b ....   → O(1)
                                                          since size = 10,
           for (int j=0; j< size; j++) → O(size)   = O(10) ~ O(1)

              b[j] = a[j] ————→   i.e. O(10) time

                                          ~ O(1) time

         delete [a] → O(1)

           a = b;   → O(1)

          size = new size; → O(1)

          a[i] = i*i; → O(1)

if   i ≠ size, then   O(1)   = O(n-1) + O(1)

if   i = size, then   O(10) ~ O(1)   = O(n)

       for (n-1) when
       ↗   i≠size   ⌐ time   complexity: O(n) ⌐

= O(n-1) + O(1) {O(1)}
        ↑   ↖ inner
      i≠size   loop → i=size