

Done C/Ver

Done C/Ver

d) `int i=2;`
`while (i < n) {`
`Print (i);`
`i = i * i;`
`}`

i	
2	0
4	1
16	2
256	3

↓

$$n = 2^{2^r}$$

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

$$\log(\log(n)) = r \log(2)$$

$$\rightarrow r = \log(\log(n))$$

b) `for (int i=1; i <= n; i++)`
`if (i % (int) sqrt(n) == 0)`
`for (int k=0; k <= i/3; k++)`
`Print (k);`

$O(n)$
 $O(n)$

$$i^3 = n^3$$

$$n^3 = n$$

\sqrt{n} times

$$1 \rightarrow \sqrt{n}$$

$$O(n) * O(\sqrt{n})$$

$$O(n^2)$$

Summation of $O(i^3) + (1 \rightarrow \sqrt{n})$

c) `for (i <= n)` $O(n)$ Sum
`for (k <= n)` $O(n)$
`if (A[k] == i)`
`for (m <= n)` $O(n)$
 $\log(n) : O(n \log n)$

d) `for (i < n)`
`if (i == size)`
`int *b = new int [new size];`
`for (j < size)`
`delete [a];`


```

d.) int f (int n) {
    int *a = new int [10];
    int size = 10;

```

$O(n)$ For (int i=0; i<n; i++) { $O(n)$

```

    if (i == size) {

```

```

        int newSize = 3 * size / 2;

```

```

        int *b = new int [newSize];

```

$O(n)$ For (j=0; j<size; j++) b[j] = a[j]; $O(n)$

```

        delete [] a;

```

```

        a = b;

```

```

        size = newSize;
    }

```

```

    a[i] = i * i;
}

```

```

}

```

→ Sum the inner and outer loop
because size is based on n

$O(n^2)$