Part (a)

```
void f1(int n)
{
    int i=2;
    while(i < n){
        /* do something that takes O(1) time */
        i = i*i;
    }
}
```

int i=2; → O(1)

while(i < n){ → O(1)

i = i*i; → O(1)

n	1	2	3	4	5....16	17...256
# iterations	0	0	1	1	2	3

iteration				
1	$i < n$	$i^{(2^{1-1})} < n$	$2^1 < n$	$2^1 = n$, $\log n = 1$
2	$i^2 < n$	$i^{(2^{2-1})} < n$	$2^2 < n$	$\log n = 2$
3	$i^4 < n$	$i^{(2^{3-1})} < n$	$2^4 < n$	$\log n = 4$
4	$i^8 < n$	$i^{(2^{4-1})} < n$	$2^8 < n$	$\log n = 8$
⋮	⋮	⋮		

iterations ↓

$\log(\log n) = \log 1 = 0$

$\log(\log n) = \log_2 2 = 1$

$\log(\log n) = \log 4 = 2$

$\log(\log n) = \log 8 = 3$

$O(\log(\log n))$

Part (b)

```
void f2(int n)
{
    for(int i=1; i <= n; i++){
        if( (i % (int)sqrt(n)) == 0){
            for(int k=0; k < pow(i,3); k++) {
              /* do something that takes O(1) time */
            }
        }
    }
}
```

$\rightarrow O(n)$

$i = n\sqrt{n}$

$\sqrt{n}$ times true

first: $(\sqrt{n})^3$ times

last: n^3 times

inside for loop

entry:	1	2	3	4	$\sqrt{n}$
	$i = 1 \cdot \sqrt{n}$	$i = 2\sqrt{n}$	$i = 3\sqrt{n}$	$i = 4\sqrt{n}$	$i = \sqrt{n} \cdot \sqrt{n}$
iteration;	$\sqrt{n}^3 = n^{\frac{3}{2}}$	$(2\sqrt{n})^3 = 8n^{\frac{3}{2}}$	$(3\sqrt{n})^3 = 27n^{\frac{3}{2}}$	$(4\sqrt{n})^3 = 256n^{\frac{3}{2}}$	$\sqrt{n}^3 \sqrt{n}^3$
					$2\sqrt{n}^3$

(k<) $\rightarrow k < (i\sqrt{n})^3 = \sqrt{n}^3 \cdot i^3$

$$\sum_{i=1}^{n} O(1) + \sqrt{n}^3 \times \sum_{i=1}^{\sqrt{n}} O(i^3)$$

$$= O(n) + n\sqrt{n}\, O(\sqrt{n}^{3+1})$$

$$= O(n) + n\sqrt{n}\, O(n^2) = O(\sqrt{n}\, n^3)$$

Part (c)

```
for(int i=1; i <= n; i++){ → O(n)
  for(int k=1; k <= n; k++){ → O(n)
    if( A[k] == i){ n
      for(int m=1; m <= n; m=m+m){
        // do something that takes O(1) time
        // Assume the contents of the A[] array are not changed
      }
    }
  }
}
```

inner-inner for loop:

entry(n):	1	2	3	4	5 15	16
# iterations:	1	2	2	3	3 3	4

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

$$= O(n^2) + O(n \log n)$$

$$= O(n^2)$$

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

Guides Player

$n=30$	size = 10	inner loop #
$i=10$	newsize = 15 $\rightarrow$	10
$i=15$	newsize = 22 $\rightarrow$	15
$i=22$	newsize = 33 $\rightarrow$	22

$$10 + 15 + 22$$

$$= 10\left(\frac{3}{2}\right)^0 + 10\left(\frac{3}{2}\right)^1 + 10\left(\frac{3}{2}\right)^2$$

$$= \sum_{k=0}^{\log_{\frac{3}{2}} \frac{n}{10}} \theta\left(10\left(\frac{3}{2}\right)^k\right) \qquad 10\left(\frac{3}{2}\right)^k < n \rightarrow k < \log_{\left(\frac{3}{2}\right)}\left(\frac{n}{10}\right)$$

$$= \theta\left(\frac{3}{2}^{\log_{\frac{3}{2}} \frac{n}{10}}\right) = \theta\left(\frac{n}{10}\right) = \theta(n)$$