Q3 a)

```
f1  int n

    int i = 2
    while (i < n)
        i = i * i
```

$$\sum_{i=2}^{?} \Theta(1)$$

f	0	1	2	3	4	
i	2	4	16	256		stop when i = n
	2^{2^0}	2^{2^1}	2^{2^2}	2^{2^3}		

$n = 2^{2^k}$

$\log\log(n) = k$

$$\sum_{k=0}^{\log\log n} \Theta(1)$$

$\Theta(\log(\log n))$

Looking at the table we can derive the pattern of 2^{2^k}.

However, the while loop does not run that amount of times given it depends on n. Therefore, the worst run time is actually $\log\log n$. By doing the inverse and going backwards, we can get k through $\log\log n$

(Q3 b)

```
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++)
                O(1)
```

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

if $n = 9$, 3 times $1 \% 3$

2

(3)

4

5

(6)

7

8

(9)

k	1	2	3	...	k	k = ?
i	$1\sqrt{n}$	$2\sqrt{n}$	$3\sqrt{n}$	...	$i=k\sqrt{n}$	stop when $i=n$

$$T(n) = O(n) + \sum_{i=1}^{\sqrt{n}} \sum_{t=0}^{i^3-1} O(1)$$

$$+ \sum_{i=1}^{\sqrt{n}} n^3$$

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

$$+ O(n^{7/2})$$

$$= O(n^{7/2})$$

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

↗ from m=m+m

outer for loops ↖

$O(n^2) + \sum_{m=1}^{n} \Theta(\log n)$

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

$= O(n^2)$

A[k] is only going to be correct once

d. 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
```

i = 0

...

i = 9

$O(1)$

i = 10

i = size

newsize = $\frac{3}{2} \cdot 10$

Second time

new size = $\left(\frac{3}{2}\right)^2 \cdot 10$

inner : 15 runs

size : 22

new size = $\left(\frac{3}{2}\right)^3 \cdot 10$

size : 33

3 times for $n = 30$

4 times for $n = 40$

$$\sum_{i=0}^{?} \left(\frac{3}{2}\right)^i \cdot 10$$

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

$$\sum_{i=0}^{n-1} O(1) + \sum_i \sum_{j=0}^{size-1} O(1)$$

$n = 21$

$i = 10$

$10 + 15 = 25$

$i = 15$

size = 22

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

(0 ↓ n; 1 ↓ n; 2 ↓ n)

times we add determined by n

$n = 30$

$$\sum_{x=0}^{n} \left(\frac{3}{2}\right)^{x} \cdot 10$$

$n = 21$, $m = 1$ $\Rightarrow \left(\frac{3}{2}\right)^0 10 + \left(\frac{3}{2}\right)^1 10$

$\vdots$

$n = 22$, $m = 1$

$n = 23$, $m = 2$ $\left(\frac{3}{2}\right)^0 10 + \left(\frac{3}{2}\right)^1 10$

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

$n = 33$, $m = 3$

$m = \log_{3/2}(n/10)$ → starting

$$\sum_{k=0} \left(\frac{3}{2}\right)^k \cdot 10$$

$n = 30$ $\log_{3/2}(n/10) = 2.7$

$n = 21$ $\log_{3/2}(21/10) = 1.82$

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

$$\sum_{k=0} \left(\frac{3}{2}\right)^k \cdot 10$$

```
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;
```

$a[i] = i*i$

$$\sum_{i=0}^{n-1} \Theta(1) + \left(\sum_{i}^{?} \sum_{j=0}^{\text{size}-1} \Theta(1) \right) \quad \left(\frac{3}{2}\right)^k \cdot 10$$

The first component to account for is the outer for loop which runs n times hence $O(n)$

But then we must see the work done by the if statement. The if statement will run a certain amount of times and that depends on the value of n. If we substitute in values for n, we can find a pattern:

size begins at $i = 10$, the next time around size is $3/2 \cdot 10$, the next time, it is $3/2 \cdot 10 \cdot 3/2$. The sizes are determined by $(3/2)^k \cdot 10$. The number of times $(3/2)^k \cdot 10$ is ran depends on n. If $n = 21$, size will be 10, then 15, meaning the for loop runs 10 + 15 times.

After finding this pattern, we need to find the upper bound, how many times the if statement will run. The max times it can run is $\log_{3/2}(n/10)$.

$$\Theta(n) \times \sum_{k=0}^{\log_{3/2}(n/10)} \left(\frac{3}{2}\right)^k \cdot 10$$

using geometric series

10. $\dfrac{\frac{3}{2}^{\log_{3/2}(\frac{n}{10})} - 1}{\frac{3}{2} - 1}$

$= 10 \cdot \dfrac{\frac{n}{10} - 1}{1/2}$

$\dfrac{n - 10}{1/2}$ $= \boxed{\Theta(n)}$