Problem 3: Runtime Analysis

input	output			
1	2	$= 2^1$		1
2	4	$= 2^2$		2
3	16	$= 2^4$	2^{k-1}	4
4	256	$= 2^8$		8
5	65536	$= 2^{16}$		
6	4bill....			

pattern $= 2^{2^{k-1}}$

Part (a)

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

$$2^{2^{k-1}} = n$$
$$\log(2^{2^{k-1}}) = \log(n)$$
$$2^{k-1} = \log(n)$$
$$2^{k-1} = \log(n)$$
$$\log(2^{k-1}) = \log(n)$$
$$k-1 = \log(\log(n))$$

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

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 */
            }
        }
    }
}
```

n times

$n^{\frac{1}{2}}$ $\sqrt{n}$ times

i^3

inner for loop

$$= \sum_{k=0}^{i^3} \Theta(1) = \Theta(i^3)$$

= If statement $= \Theta(\sqrt{n})$

= outer for loop $= \sum_{i=1}^{n} (\Theta(1) + \Theta(i^3)) = \Theta(n) + \sum_{i=1}^{n} \Theta(i^3)$

from the slideshow

K	1	2	3	...	arbitrary k	Stop when $k = \sqrt{n}$
i	$\sqrt{n}$	$2\sqrt{n}$	$3\sqrt{n}$	...	$i = k\sqrt{n}$	Stop when $i = n$

$$n + n^{\frac{1}{2}} + n^{4} = n^{\frac{3}{2}}$$
$$\Theta(n^{\frac{3}{2}})$$

$$= \Theta(n) + \sum_{k=1}^{\sqrt{n}} (k\sqrt{n})^3$$
$$= \Theta(n) + \sum_{k=1}^{\sqrt{n}} k^3 n\sqrt{n}$$

$$\sum_{i=1}^{n} \Theta(i^p) = \Theta(n^{p+1})$$

$$= \Theta(n) + n\sqrt{n} \sum_{k=1}^{\sqrt{n}} k^3$$
$$= \Theta(n) + n\sqrt{n} \sum_{k=1}^{\sqrt{n}} = n^{3+1}$$
$$= \Theta(n) + n\sqrt{n}\left[(\sqrt{n})^4\right]$$
$$= \Theta(n) + n\sqrt{n}\,[n^2]$$
$$= \Theta(n) + n^{\frac{2}{2}} + n^{\frac{1}{2}} + n^{\frac{4}{2}}$$
$$= \Theta(n) + \Theta(n^{\frac{7}{2}})$$
$$= \boxed{\Theta(n^{\frac{7}{2}})}$$

Part (c)

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

will be the at most n times but should be checked n² times

$m = 1, 2, 4, 8, 16, 32, 2^k$

$2^k = n$

$k = \log n$

Assume A satisfies when i=1 so k loops through like this

$A[k] = i \Rightarrow A[1]=1 \quad A[2]=1$

$A[3]=1 \quad A[\ldots k]=1$

inner for loop executes when k=n times due to the fact we assume that the contents of the A[] array does not change

$$= \sum_{i=1}^{n} \sum_{k=1}^{n} \Theta(1) + \sum_{m=1,2,4,8}^{n} \Theta(1)$$

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

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

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

Part (d)

Notice that this code is very similar to what will happen if you keep inserting into an ArrayList (e.g. vector). Notice that this is NOT an example of amortized analysis because you are only analyzing 1 call to the function f(). If you have discussed amortized analysis, realize that does NOT apply here since amortized analysis applies to multiple calls to a function. But you may use similar ideas/approaches as amortized analysis to analyze this runtime. If you have NOT discussed amortized analysis, simply ignore it's mention.

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

$i = 10 \times \frac{3}{2}^S$

$n = 10 \times \frac{3}{2}^S$

$S = \log_{\frac{3}{2}} \frac{n}{10}$

$$= \sum_{i=0}^{n} \left(\Theta(1) + O\left(\sum_{j=1}^{10 \times \frac{3}{2}^k} \Theta(1) \right) \right)$$

$$= \sum_{i=0}^{n} \Theta(1) + \sum i \sum_{j=1}^{10 \times \frac{3}{2}^k} \Theta(1)$$

$$= \Theta(n) + \sum_{k=0}^{\log_{\frac{3}{2}} \frac{n}{10}} \sum_{j=0}^{10 \times \frac{3}{2}^k} \Theta(1)$$

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

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

$\sum_{i=0}^{n} c^i = \frac{c^{n+1}-1}{c-1} = \Theta(c^n)$

$n = \log_{\frac{3}{2}} \frac{n}{10}$

$$= \Theta(n) + \Theta\left(10\left(\frac{n}{10}\right)\right)$$

$$= \Theta(2n)$$

$$\boxed{= \Theta(n)}$$