Almendra Gutierrez
CSCI 104 - HW 1
Runtime Analysis

Part (a):

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

* Could be re-written as:

for(int i = 2; i<n; i = i*i)

* $\Theta(2) + \Theta(4) + \Theta(16) + \Theta(256) + \dots + \Theta(n)$

$\therefore$ Iter 0 $i = 2^1$
Iter 1 $i = 4 = 2^2$
Iter 2 $i = 16 = 2^4$
Iter 3 $i = 256 = 2^8$
Iter 4 $i = 65{,}536 = 2^{16}$
$\vdots$
Iter K $i = 2^{2^K} = n$

$\therefore$ log

$$\log_2 2^{2^K} = \log_2(n)$$

$$2^K = \log_2(n)$$

contd:

$$\log_2 2^K = \log_2 \log_2 (n)$$

$$K = \log_2 \log_2 (n)$$

$$* \sum_{K=0}^{\log_2 \log_2 (n)} \Theta\left(2^{2^K}\right) = \Theta \sum_{K=0}^{\log \log n} 2^{2^K}$$

$$= \Theta\left(2^{2^{\log_2 \log_2 (n)}}\right)$$

$$= \Theta\left(2^{\log_2 (n)}\right)$$

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

Part (b):

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

for (int j=0; j<pow(i,3); j++

$$\star\ T_i(n) = \sum_{j=0}^{i^3-1} \Theta(1) = \Theta(i^3)$$

if (i % $\sqrt{n}$ == 0)

→ When $i \% \sqrt{n} \neq 0$, takes:

$$\Theta\left(\sum_{j=0}^{i^3-1} \Theta(1)\right)$$ upper bound

$$\Omega\left(\sum_{j=0}^{i-1} \Theta(1)\right)$$ lower bound

→ Input that will execute loop every time: $i \% \sqrt{n} == 0$

for(int i=1; i<=n; i++)

$$\star\ T_i'(n) = \sum_{i=0}^{n-1} \Theta(\Theta(i^3))$$

$$= \Theta \sum_{i=0}^{n-1} i^3 = \frac{n^2(n+1)^2}{4}$$

$$\boxed{\text{Runtime} = \Theta(n^4)}$$

Part (c)

```
for ( int i=1 ; i<=n ; i++ )
  for ( int K=1 ; K<=n ; K++ )
    if ( A[K] == i )
      for ( int m=1 ; m<=n ; m=m+m )
        // do something O(1)
```

```
for ( int m=1 ; m<=n ; m=m+m )
```

$\rightarrow \Theta(1)+\Theta(2)+\Theta(4)+\Theta(8)+\Theta(16)+\dots\Theta(n)$

Iter 0 $\quad i=1=2^0$
Iter 1 $\quad i=2=2^1$
Iter 2 $\quad i=2^2$
Iter 3 $\quad i=2^3$
$\vdots \qquad \vdots$
Iter K $\quad i=2^k=n$

$\rightarrow \log_2 2^k = \log_2 n$

$K = \log_2(n)$

$\rightarrow T_i'(n) = \sum_{m=1}^{\log_2(n)} 2^k = \Theta \sum_{m=1}^{\log_2 n} 2^k$

$= \Theta\left(2^{\log_2 n}\right) = \Theta(n)$

```
if ( A[K] == i )
```

$\rightarrow T_i'(n) = \Theta(\Theta(n))$

```
for(int K=1; K<=n; K++)
```

$$T_i'(n) = \sum_{K=1}^{n} \Theta(\Theta(n)) = \Theta \sum_{K=1}^{n} n$$

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

```
for(int i=1 ; i<=n ; i++)
```

$$T_i'(n) = \sum_{i=1}^{n} \Theta(n^2)$$

$$\boxed{\text{Runtime} = \Theta(n^3)}$$

Part (d)

```
int f (int n)
{
   int *a = new int [10];      } Θ(1)
   int size = 10;
(3) for (int i = 0; i < n; i ++)
      {
    (2) if (i == size)
         {
            int newsize = 3*size/2;      } Θ(1)
            int *b = new int [newsize];
        (1) for (int j = 0; j < size; j ++) b[j] = a[j];   -- Θ(1)
            delete [] a;
            a = b;                   } Θ(1)
            size = newsize;
         }
         a[i] = i*i;  --> Θ(1)
      }
}
```

for (int j = 0; j < size; j++) (1)

$$T_i'(n) = \sum_{j=0}^{size-1} \Theta(1) = \Theta(size)$$

if (i = = size)

$$T_i'(n) = \Theta(\Theta(size))$$

for(int i = 0; i<n; i++)

$$T_i'(n) = \sum_{i=0}^{n-1} \Theta(\Theta(size))$$

$$size = 10$$

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

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