<HW1>

- Q3. Runtime Analysis
  - part(a)

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

$\rightarrow \Theta(1)$

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

Runtime of $f1(n)$ =

$T(n) = \Theta(1) + \sum_i (\Theta(1))$

$\Rightarrow$ If $n = 17$, while loop will be repeated 3 times

for $i = 2, 4, 16$

if $n = 16$, the loop will be repeated 2 times

for $i = 2, 4$

if $n = 5$ the loop will be " 2 times

for $i = 2, 4$

if $n = 4$, the loop will be " 1 time

for $i = 2$.

if $n = 3$, " 1 time

for $i = 2$

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
k ; repeat	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3
max i :	2		2, 4												2, 4, 16

k & i :

$n = 5 \sim 16$	$n = 17 \sim 258$	$n = 257 \sim 2^{16}$	$\sim 2^{32}$	$\sim 2^{64}$	$n = 2 \sim 4$
$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 1$
$i = 4 = 2^2 = 2^{(2^1)}$	$i = 16 = 2^4 = 2^{(2^2)}$	$i = 256 = 2^8 = 2^{(2^3)}$	$i = 2^{16} = 2^{(2^4)}$	$i = 2^{32} = 2^{(2^5)}$	$i = 2$

$\boxed{i = 2^{(2^{k-1})}}$ $\Rightarrow$ $\boxed{k = \left(\log_2 \left(\log_2 (i)\right)\right) + 1}$ $(k \geq 1)$

$i < n \Rightarrow k < \log_2(\log_2(n)) + 1$

$\Rightarrow T(n) = \Theta(1) + \sum_{k=1}^{\log_2(\log_2 n)} \Theta(1)$

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

Part(b)

```c
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 */
            }
        }
    }
}
```

$$T(n) = \sum_{i=1}^{n} \Big(\underbrace{\theta(1)}_{\text{if-else statement}} + \sum_{k=0}^{i^3-1} \theta(1) \Big)$$

$$= \theta(n) + \boxed{\sum_i} \sum_{k=0}^{i^3-1} \theta(1)$$

when $n = 1$, for $i = 1$, inner loop is excuted on $k = 0$

4 — 2, 4

9 — 3, 6, 9

0 ~ 7

0 ~ 63

0 ~ 26

0 ~ 215

0 ~ 728

α	1	2	3	4		α		$\sqrt{n}$
i	$\sqrt{n}$	$2\sqrt{n}$	$3\sqrt{n}$	$4\sqrt{n}$	...	$\alpha\sqrt{n}$	...	n

$i = \alpha\sqrt{n}$, $i \leq n \rightarrow \alpha \leq \sqrt{n}$

$$= \theta(n) + \sum_{\alpha=1}^{\sqrt{n}} \sum_{k=0}^{(\alpha\sqrt{n})^3-1} \theta(1)$$

$$= \theta(n) + \sum_{\alpha=1}^{\sqrt{n}} \theta\big((\alpha\sqrt{n})^3 - 1 \big)$$

$$= \theta(n) + \theta\Big(n\sqrt{n} \sum_{\alpha=1}^{\sqrt{n}} \alpha^3 \Big) - \sum_{\alpha=1}^{\sqrt{n}} \theta(1)$$

$$= \theta(n) - \theta(\sqrt{n}) + \theta\big(n\sqrt{n} \cdot (\sqrt{n})^4 \big) = \theta(n^{\frac{7}{2}})$$

part(C)

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 that takes O(1) time
        // Assume the contents of the A[] array are not changed
      }
    }
  }
}
```

$$T(n) = \sum_{i=1}^{n}\left(\sum_{k=1}^{n}\left(\underbrace{\theta(1)}_{\text{if Statement}} + \sum_m \theta(1)\right)\right)$$

$$= \theta(n^2) + \sum_{i=1}^{n}\left(\sum_{k=1}^{n} \sum_m \theta(1)\right)$$

$$= \theta(n^2) + \sum_{j=1}^{n}\left(\sum_{m=1}^{\lceil \frac{n}{t} \rceil} \theta(1)\right)$$

$$= \theta(n^2) + \sum_{j=1}^{n} \theta\left(\left\lceil \frac{n}{t} \right\rceil\right)$$

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

worst case of if-statement is for all K, $1 \le A[K] \le n$.
Let's assume that every A[K] are different
Specifically we can say A[K]=K.
i.e., K=i. This assumption is correct because the $\boxed{\text{inner part}}$ is independent from the outside variables.

So, in this case, there's always one time when if stat. is true for the second for loop.

For the third loop, we can set $m < \frac{n}{t}$

part (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;
    }
}
```

$\theta(1)$

if $10 \geq n$: Best case : $\theta(n)$

else $10 < n$: Worst Case :

↳ When $i=10 \rightarrow$ newsize = 15.

b = new int [15]

$\sum_{j=0}^{10} \theta(1)$

a = int[15]

size = 15.

↳ $i=15 \rightarrow$ newsize = 21

b = int[21] = a

$\sum_{j=0}^{15} \theta(1)$

size = 21

$10 \times \frac{3}{2} = 15$

$15 \times \frac{3}{2} = 21$

$21 \times \frac{3}{2} = 30$

$30 \times \frac{3}{2} = 45$

$\vdots$

$$\Rightarrow T(n) = \theta(1) + \sum_{i=0}^{n-1} \theta(1) + \sum_i \left(\theta(1) + \sum_{j=0}^{size} \theta(1) \right)$$

$$= \theta(n) + \sum_i \left(\theta(1) + \sum_{j=0}^{10 \cdot (\frac{3}{2})^k} \theta(1) \right)$$

$10(\frac{3}{2})^k < n$

$k < \log_{\frac{3}{2}}(\frac{n}{10})$

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

$$= \theta(n) + \theta(\log n) + \sum_{k=0}^{\log_{\frac{3}{2}}(\frac{n}{10})} \left(\theta\left(10 \cdot (\tfrac{3}{2})^k \right) \right)$$

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

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

$$= \theta(n)$$