<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 $f1(n) =$

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

⇒ 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~~ 〃 → 2, 4 → 〃 → 2, 4, 16

k & i :

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

$\Rightarrow i = 2^{(2^{k-1})}$ $(k > 2)$

k & 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 */
            }
        }
    }
}
```

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

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

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

4 — 2: $0 \sim 7$; 4: $0 \sim 63$

9 — 3: $0 \sim 26$; 6: $0 \sim 215$; 9: $0 \sim 728$

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

$i = \alpha\sqrt{n}$, $\sim i = n$

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

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

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

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

Part (c)

```
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
      }
    }
  }
}
```

part(C)

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

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

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

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

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

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 is no wrong because –
$\boxed{\text{안의 내용이 영향 X}}$

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 $M = 1 \sim \left\lceil \frac{n}{t} \right\rceil$ where

geometric summation

$$10\left(\tfrac{3}{2}\right)^k < n$$

$$k < \underbrace{\left\lceil \log_{\frac{3}{2}}\left(\frac{n}{10}\right) \right\rceil}_{t}$$

```
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;
  }
}
```

$\theta(1)$

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

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

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

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

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

$$= \theta(n) + \theta(t) + 10\,\theta\left(\tfrac{3}{2}^t\right)$$

$$= \theta(n) + \theta(\log n)$$

$$= \theta(n + \log n) \;(X) = \theta(n)$$

$$\sum_{k=0}^{\lceil \log_{\frac{3}{2}}(\frac{n}{10}) \rceil} \sum_{j=0}^{k} \left(10\left(\tfrac{3}{2}\right)^j\right)$$

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

$$= \theta\left(\tfrac{3}{2}^n\right)$$

$$\sum_{k}^{\log} \tfrac{3}{2}^k$$

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

$$= \frac{n}{10}$$

$\theta(n)$

if $\underbrace{10 \ge n}$: Best case : $\theta(n)$

else $\underbrace{10 < n}$: Worst Case :

↳ When $i = 10$ → newsize = 15.
b = new int [15]
$\sum_{j=0}^{10} \theta(1)$
a = int[15]
size = 15.

↳ $i = 15$ → newsize = 21
b = int [21] = a
$\sum_{j=0}^{15} \theta(1)$
size = 21

$10 \times \frac{3}{2} = 15$
$15 \times \frac{3}{2} = 21$
$21 \times \frac{3}{2} = 30$
$30 \times \frac{3}{2} = 45$
⋮

$$\boxed{t = \log_{\frac{3}{2}}\left(\frac{n}{10}\right) \qquad \left(\frac{3}{2}\right)^t = \left(\frac{n}{10}\right) \Leftarrow \lim_{k \to \infty} 10 \times \left(\frac{3}{2}\right)^k = n}$$

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

$\rightarrow \Theta(1)$

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

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

$$= \sum_{i} \Theta(1)$$

$$\boxed{\boxed{i = 2^{(2^{k-1})} = n-1} \quad K = \log_2(\log_2(n-1)) + 1}$$

$$= \sum_{k=1}^{\lfloor \log_2(\log_2(n-1)) \rfloor + 1} \Theta(1) = \log(\log(n))$$

when $n=16$ $\lfloor \log_2(\underbrace{\log_2 15}_{3.xx}) \rfloor = 1$ $\quad k=1 \sim 2$

$\approx [1.xx]$

$n=17. \rightarrow$ $K = 1 \sim 3$