Problem 3

a.

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

2, 4, 16 ... 2^{2^h}

i increases exponentially

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

$2 \to 2+2 \quad 4 \to 4+4 \to 8 \to 8+8$

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 */
            }
        }
    }
}
```

$\sum_{i=1}^{n} \left(O\left(\sum_{i=1}^{\sqrt{n}} \sum_{k=0}^{(i\sqrt{n})^3 - 1} \Theta(1) \right) \right) = \Theta(n^{\frac{3}{2}})$

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
      }
    }
  }
}
```

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

j	m
1	1
2	2
3	4
4	8
5	16
6	32
7	64

$\sum_{m=1}^{n} \frac{1}{m} \to \log(n)$

$\sum_{i=1}^{n} \left(\sum_{k=1}^{n} + \sum_{m=1}^{n} \log(k) \right)$

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

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;
     }
}
```

$10 \cdot \left(\frac{3}{2}\right)^h < n \quad \left(\frac{3}{2}\right)^h < \frac{n}{10}$

$\sum_{i=0}^{h-1} + \sum_{i=0}^{\log_{\frac{3}{2}}\left(\frac{n}{10}\right)} \sum_{j=0}^{size-1} \Theta(1)$

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

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

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

$10 \cdot \frac{n}{10}$