HQ1 CSE 222 Report

```
private static int searchSub(String small, String big, int index, int n){
    if(big.length() < small.length()){
        return -1;                                                        1
    }                                                                     1
    else if (big.indexOf(small) != -1 && index-- == 1){                   1
        return n;                                                         1
    }                                                                     1
    else
        return searchSub(small, big.substring(big.indexOf(small)+1), index,     1
    n+big.indexOf(small)+1);
}                                                                         T(n-1)
```

$T(n) = T(n-1) + 1$

Best case $T(1) = 1$

1

worst case $\Theta(n)$ **Best case** $\Theta(1)$

$\cancel{T(n+1) = T(n) + 1}$
$\cancel{T(n+2) = T(n+1) + 1}$
$T(n+3) = T(n+2) + 1$

$T(n+k) = T(n) + k$

$T(n) = T(n+k) - k \leftarrow k = 1-n$ $\quad n+k = 1$

$T(n) = T(1) + n - 1$
$T(n) = n$

#Q2

```
    Public Static Int Q2( Int[] arr, int index, int first, int last
int findex, int Sindex ){                                          1
  if( arr[index] == last ){                                        1
      Sindex = index;                                              1
      return Sindex - findex - 1;                                  1
  }
  else if ( index == arr.length - 1) {                             1
    Sindex = arr.length - 1;                                       1
    if ( arr[index] < last ){                                      1
          return Sindex - findex;                                  1
    }                                                              1
    return 0;                                                      1
  }
  else if ( arr[index] == first) {                                  1
     return Q2( arr, index + 1, first, last, index, Sindex);     T(n-1)
  }
else if ( arr[index] < first && first < arr[index+1] ){             1
   return Q2( arr, index+1, first, last, index, Sindex);         T(n-1)
  }
else if( arr[0] > first) {                                          1
   return Q2( arr, index+1, first, last-1, Sindex);              T(n-1)
else
   return Q2 ( arr, index+1, first, last, findex, Sindex);          1
}                                                                T(n-1)
```

$T(1) = 1$

$T(n) = T(n-1) + 3$

$T(n) = n + 3$

Best case

$\Omega(1)$

worst case

$O(n)$

#03

#04

```
foo(integer1, integer2)
    if(integer1 < 10) or (integer2 < 10)
        return integer1 * integer2                                              1
                                                                                1
    n = max(number_of_digits(integer1), number_of_digits(integer2))            n
    half = int(n/2)                                                            1
    int1, int2 = split_integer(integer1, half)                                 log n
    int3, int4 = split_integer(integer2, half)                                 log n
    sub0 = foo(int2, int4)
    sub1 = foo((int2+int4), (int1+int3))
    sub2 = foo(int1, int3)                                                     ] n^2
    return (sub2 * 10^(2*half)) + ((sub1-sub2-sub0)*10^(
    half)) + sub0)
```

$T(0) = 0$

$T(n) = 3T(n/(n^2)^2 +) \Theta(1)$

$T(n) = 3^2(T(n/(n^2)^2) + \Theta(1)$

$T(n-k) = 3^{\log(\log k)} T(0) + \Theta(1) \Rightarrow T(n) = \Theta(\log n)$

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