18 October 2011

451 Algorithm Analysis
Assignment #2
Return date: 25 October 2011 09:30

(20 pts. Q1) The algorithm *Secret* is a recursive algorithm. Write an iterative version of the same algorithm.

```
ALGORITHM  Secret (n)
  // Input : n : int
  // Output : .....
  if  n = 1
    return  1
  else
    return  Secret(n − 1) + 2 * n − 1
```

(20 pts. Q2) How many times is the basic operation executed? Establish your formula and show your solution. What is the value of r? What is the efficiency of this class?

```
ALGORITHM  Mystery (int  n)
  r ← 0
  for  i ← 1  to  n − 1  do
    for  j ← i + 1  to  n  do
      for  k ← 1  to  j  do
        r = r + 1
  return  r
```

(60 pts. Q3) Traditional binary search algorithm splits the input into two two sets of almost-equal sizes.

(25 pts) a) Similarly, write a recursive, divide-and-conquer search algorithm (as pseudocode) which splits the sorted input into two sets of sizes approximately one-third and two-thirds.

(10 pts) b) Set up a recurrence for the number of key comparisons assuming that the algorithm goes to the one-third of the input (left of the input). Explain how you set up the recurrence. (You may assume that $n = 3^k$)

(15 pts) c) Solve the recurrence for $n = 3^k$ by backward substitution.

(10 pts) d) Compare this algorithm's efficiency with that of traditional binary search. (In traditional binary search, the number of key comparisons is $\log_2 n$).