

## 2) (10 pts) ANL (Algorithm Analysis)

a) (5 pts) Given that a function has time complexity  $O(n^2)$ , if the function takes 338 ms for an input of size 13000, how long will the same function take for an input of size 8000?

$$T(n) = cn^2$$

$$T(13000) = c13000^2 = 338ms$$

$$c = \frac{338}{13^2} \times \frac{1}{10^6} ms = \frac{338}{169} \times 10^{-6} ms = 2 \times 10^{-6} ms$$

$$T(8000) = c(8000)^2 = (2 \times 10^{-6} ms) \times 8^2 \times 10^6 = 128ms$$

128 ms

**Grading: 2 pts for computing the constant c, 2 pts substituting 8 into general equation to find time for input size of 8000, 1 pt for simplifying the final answer.**

b) (5 pts) What is the run-time of the segment of code below, in terms of the variables  $n$  and  $k$ ? Please provide a Big-Oh bound and briefly justify your answer. (Assume  $k$  has already been defined as set to a value prior to the code segment shown.)

```
int i, total = 0;
for (i=0; i<n; i+=2) {
    int start = k;
    while (start > 0) {
        total += ((k|i) & start);
        start /= 2;
    }
}
```

The outer loop runs  $n/2$  times. The inner loop will always run  $O(\lg k)$  times, since  $k$  never changes during the code segment and each iteration of the while loop divides start by 2. Since the two loops are independent of each other in terms of number of times they run, it follows that the run time of the code segment is  $O(n \lg k)$ .

$O(n \lg k)$

**Grading: 2 pts outer loop analysis, 2 pts inner loop analysis, 1 pt final answer**