

COP 3502 Study Group Sheet: Algorithm Analysis

Directions: Work together as a group to try to solve these problems. Talk through issues and see if you can convince yourselves of the right path to move forward. In groups with a TA/ULA, towards the end of the session some of the solutions will be covered. At the end of the week, the solutions will be posted for everyone.

Directions: For questions 1 – 4, utilize the technique shown in class of setting up an equation with a constant, solve for that constant, and then answer the given question.

- 1) For an $O(n^3)$ algorithm, one data set with $n = 3$ takes 54 seconds. How long will it take for a data set with $n = 5$?
- 2) For an $O(2^n)$ algorithm, a friend tells you that it took 17 seconds to run on her data set on a $O(2^n)$ algorithm. You run the same program, on the same machine, and your data set with $n = 7$ takes 68 seconds. What size was her data set?
- 3) For an $O(N^k)$ algorithm, where k is a positive integer, an instance of size M takes 32 seconds to run. Suppose you run an instance of size $2M$ and find that it takes 512 seconds to run. What is the value of k ?
- 4) Assume that an $O(\log_2 N)$ algorithm runs for 10 milliseconds when the input size (N) is 32. What input size makes the algorithm run for 14 milliseconds?

Directions: For questions 5 – 9, represented as functions with appropriate names, determine the run-time for the function in terms of the variable n . The answers should simply be Big-Oh answers, but you need to provide ample justification for your answers. You may assume that n is a positive integer.

```
int function5(int A[], int B[], int n) {  
  
    int i, j, sum = 0;  
    for (i=0; i<n; i++)  
        for (j=0; j<n; j++)  
            if (A[i] == B[j])  
                sum++;  
    return sum;  
}
```

```

int function6(int A[], int B[], int n) {
    int i=0,j=0;

    while (i < n) {
        while (j < n && A[i] > B[j]) j++;
        i++;
    }
    return j;
}

```

```

int function7(int A[], int B[], int n) {
    int i=0,j;

    while (i < n) {
        j=0;
        while (j < n && A[i] > B[j]) j++;
        i++;
    }
    return j;
}

```

```

void function8(int n) {

    while (n > 0) {
        printf("%d\n", n);
        n = n/2;
    }
}

```

```

int function9(int n) {
    int i,j;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            if (j == 1)
                break;
    return j;
}

```