

Computer Science 311

Recitation 1

Weisi Fan

02-01-2021

1 ASYMPTOTIC NOTATION

O -notation:

$f(n)$ is $O(g(n))$ if and only if there exist positive constants c and n_0 such that

$$f(n) \leq c * g(n) \text{ for all } n > n_0$$

Equivalently, $f(n) = O(g(n))$ if and only if there exists a constant $c \geq 0$ such that

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq c$$

Give the asymptotic growth rate using Big-O for each of the following functions.

$$\begin{array}{lll} f_1(n) = 4n \log n + 2n & f_2(n) = 2^{10} & f_3(n) = 2 \log n \\ f_4(n) = 3n + 100 \log n & f_5(n) = 4n & f_6(n) = 2^n \\ f_7(n) = n^3 + 10n & f_8(n) = n^3 + 5n^2 & f_9(n) = n \log n \end{array}$$

2 ALGORITHM ANALYSIS

Give big-O bounds, in terms of n , for the worst-case time of each of the algorithms below.

(a)

Algorithm 1: AlgA(A)

Input: An array A

```

1  $n = A.length$ 
2  $prefix = 0$ 
3  $total = 0$ 
4 for  $i = 0$  to  $n - 1$  do
5   |  $prefix = prefix + A[i]$ 
6   |  $total = total + prefix$ 
7 return  $total$ 
```

(b)

Algorithm 2: AlgB(A, B)

Input: Array A and B with same length

```

1  $n = A.length$ 
2  $count = 0$ 
3  $total[0,1,...,n-1] = \{0, 0, ..., 0\}$ 
4 for  $j = 0$  to  $n - 1$  do
5   | for  $k = 0$  to  $j$  do
6   | |  $total[j] = total[j] + A[k]$ 
7 return  $total[0,1,...,n-1]$ 
```

(c)

Algorithm 3: AlgC(A, B)

Input: Array A and B with same length

```
1  $n = A.length$ 
2  $count = 0$ 
3 for  $i = 0$  to  $n - 1$  do
4    $total = 0$ 
5   for  $j = 0$  to  $n - 1$  do
6     for  $k = 0$  to  $j$  do
7        $total = total + A[k]$ 
8       if  $B[i] == total$  then
9          $count = count + 1$ 
10 return  $count$ 
```
