

AUTHOR1: Austin Welch austinmw@bu.edu

AUTHOR2: Dave Ferullo dferullo@bu.edu

AUTHOR3: Wasim Khan wasimk95@bu.edu

% EC 504 - Advanced Data Structures % Spring 2016 % Homework 2, Due Feb
28 (11pm)

Asymptotics, 25 pts

Place the following functions from asymptotically smallest to largest. When two functions have the same asymptotic order, put an equal sign between them. Provide an explanation for your ordering.

$$1, n^3, n^{n^n}, n^{\frac{1}{n}}, 0, \frac{n}{4}, n^9 + n + 2, \sqrt[3]{n}, (n+1)^n, \sum_{k=1}^{\log n} \frac{n}{3^k}, \left(1 + \frac{1}{n}\right)^n, \prod_{k=1}^n \left(1 - \frac{1}{k^3}\right), \log n$$

Answer:

$$0 = \prod_{k=1}^n \left(1 - \frac{1}{k^3}\right), 1, n^{\frac{1}{n}}, \left(1 + \frac{1}{n}\right)^n, \log n, \sqrt[3]{n}, \frac{n}{4}, \sum_{k=1}^{\log n} \frac{n}{3^k}, n^3, n^9 + n + 2, (n+1)^n, n^{n^n}$$

Solved using Python program taking N as 1000 and then sorting.

Recurrences, 25 pts

For each of the following functions, provide:

1. A recurrence $T(n)$ that describes the worst-case runtime of the function in terms of n as provided (i.e. without any optimizations)
2. The tightest asymptotic upper and lower bounds you can find for $T(n)$

““ def A(n): if (n == 0): return 1 else: return A(n-1) * A(n-1) * A(n-1)

Answer: 3^n

def B(n): if (n == 0): return 1 if (B(n//2) >= 5): return B(n//2) + 5 else:
return 5

Answer: n

def C(n): if (n <= 1): return 1 sum=0 for ii in range(int(math.sqrt(n))): sum
+= C(int(math.sqrt(n))) return sum

Answer: $\log(\log(n))$

def D(n): if (n <= 1): return 1 count = 1 tmp = D(n//2) for jj in range(n):
ii=1; while (ii<n): if (tmp < math.exp(ii+jj)): count += 1 ii+=2 return D(n//2)
(count % 2)

Answer: $n \log^2(n)$

def E(n): if (n == 0): return 1 if (n == 1): return 3 return E(n-1) + E(n-
2)*E(n-2)

Answer: 2^n