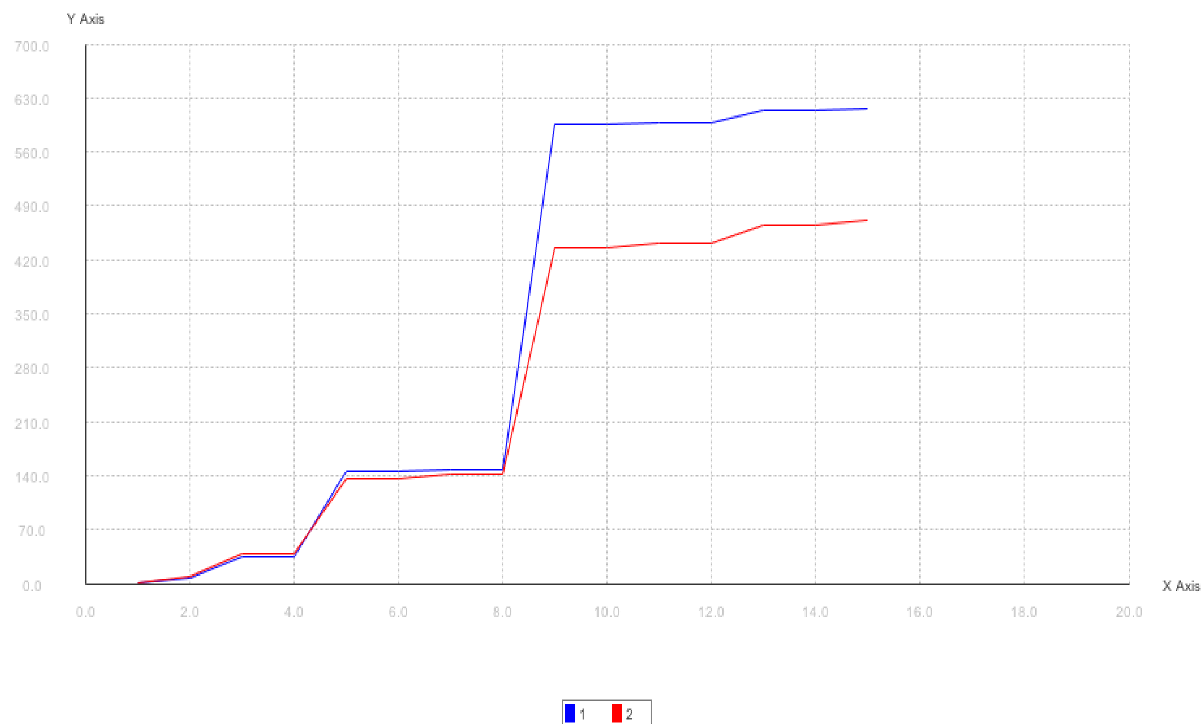
COMP251 Assignment 4

Lucia Berger — 260473661

Sources: Stack Overflow, Wikipedia, Class Slides

1 Problem One



Karatsuba Analysis The Karatsuba algorithm becomes more efficient after at size 5. At size 4, the results are 34,39 where there after at 5, the results are 145,135 and this pattern continues there on.

2 Problem Two

Master Theorem Cases are defined in homework sheet class notes:

a) By Case I:

$$T(n) = 25 * T(n/5) + n$$
$$a = 25b = 5k = log_{25}5$$
$$f(n) = O(n) = O(n^c)$$
$$c < 2 < k$$
$$O(n^2)$$

b) By Case III:

$$T(n) = 2 * T(n/3) + n * log(n)$$
$$a = 2b = 3k = log_3 2 = 0.63$$
$$f(n) = O(nlogn) = \delta(n^c)c = 2$$
$$c > k$$
$$\alpha f(n/b)\alpha f(n)$$
$$2(nlogn/3log3) <= nlogn$$
$$\alpha = (2/3log3) < 1$$
$$O(nlogn)$$

c) By Case II:

$$T(n) = T(3n/4) + 1$$
$$a = 1b = 3/4$$
$$f(n) = O(1) = O(n^k log^p n)$$
$$k = 0p = 0O(n^o log^o * n) = O(1)$$
$$T(n) = 0(n^k log^p + 1n)$$
$$O(logn)$$

d)By Case III:

$$T(n) = 7 * T(n/3) + n^3$$
$$a = 7b = 3k = log_3 7 = 1.77$$
$$f(n) = n^3 = O(n^3)c = 3$$
$$\delta(n^3)c > k$$
$$\alpha f(n/6) <= \alpha f(n)$$
$$7(n3/27) <= \alpha n^3 \alpha = 7/27$$
$$T(n) = O(f(n)) = 0(n^3)$$
$$0(n^3)$$

e) None of the above cases apply.If we try case III, it violates regularity conditions that are required.

$$T(n/2) + n(2 - cosn)$$

3 Problem Three

Master Theorem Comparison

$$T_A(n) = 7T_A(n/2) + n^2$$

$$T_B(n) = \alpha T_B(n/4) + n^2$$

$$T_A(n) = 7T_A(n/2) + n^2$$

$$a = 7b = 2k = log_2 7 = 2.8$$

$$f(n) = n^2 = O(n^c)$$

$$c = 2c < k$$

$$T(n) = O(n^l og27)$$

Lets consider T_B:

$$T_B(n) = \alpha T_B(n/4) + n^2$$

$$k = log_f \alpha log_4 \alpha = log_2 7$$

$$\alpha = 49$$

We will be at the asymptotically lowest integer. So we are at approximately 48. $\alpha = 48$

$$f(n) = n^2 = O(n^c)c = 2$$

$$T(n) = O(n^k)$$

$$T_B(n) =)(n^l og4, 48) = /alpha = 48$$

Thus that

$$n^2.8 > n^2.79, aprox48.5, 48(integer)$$

4 Problem Four

Aggregate Analysis Let k_1 be the cost of the i operation.

Let's set the rule below:

$$k_1[k_1 mod 2 = 0] \to i$$

$$k_1[k_1 mod 2! = 0] \to 1$$

$$Operation|Cost$$

1	1
2	2
3	1
4	4
5	1

N operation must cost:

$$\sum_{i=1}^{n} k_1 <= n + \sum_{j=0}^{lgn} 2^j = n + (2n-1) < 3n$$

Thus we have, average cost of operations = total cost (3n)/number of operations(n)¡4.

By aggregate analysis, the amortized cost of operations is O(1).