Conor Roberts #1056167
January 29th, 2021
CIS3490

1.

1.1. This equation evaluates to a constant result, therefore $(n^2+1)^{10} \in \theta\ (n^{20})$

1.1.1. $(n^2+1)^{10} \approx (n^2)^{10} \approx n^{20} \in \theta\ (n^{20})$

1.1.2. $\lim\limits_{n\to\infty} \frac{(n^2+1)^{10}}{n^{20}}$

1.1.3. $\lim\limits_{n\to\infty} \frac{(n^2+1)^{10}}{(n^2)^{10}}$

1.1.4. $\lim\limits_{n\to\infty} (\frac{n^2+1}{n^2})^{10}$

1.1.5. $\lim\limits_{n\to\infty} (\frac{n^2}{n^2} + \frac{1}{n^2})^{10}$

1.1.6. $\lim\limits_{n\to\infty} (1 + \frac{1}{n^2})^{10}$

1.1.7. $= 1^{10}$

1.1.8. $=1$

1.2. This equation evaluates to a constant result, therefore $\sqrt{10n^2+7n+3} \in \theta\ (n)$

1.2.1. $\sqrt{10n^2+7n+3} \approx (n^2)^{1/2} \in \theta\ (n)$

1.2.2. $\lim\limits_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{n}$

1.2.3. $\lim\limits_{n\to\infty} \frac{\sqrt{10n^2+7n+3}}{\sqrt{n^2}}$

1.2.4. $\lim\limits_{n\to\infty} \sqrt{\frac{10n^2+7n+3}{n^2}}$

1.2.5. $\lim\limits_{n\to\infty} \sqrt{\frac{10n^2}{n^2} + \frac{7n}{n^2} + \frac{3}{n^2}}$

1.2.6. $\lim\limits_{n\to\infty} \sqrt{10 + \frac{7}{n} + \frac{3}{n^2}}$

1.2.7. $\sqrt{10}$

1.3. This equation evaluates to a constant result, therefore $2n * log(n+2)^2 + (n+2)^2 \in \theta\ n^2 log(n)$

1.3.1. $2n * log(n+2)^2 + (n+2)^2 * lg(\frac{n}{2}) \approx 4n * log(n+2) + log(\frac{n}{2})^{(n+2)^2} \approx n^2 log(n)$

1.3.2. $\lim_{n\to\infty} \frac{2n*log(n+2)^2+(n+2)^2*log(\frac{n}{2})}{n^2*log(n)}$

1.3.3. $\lim_{n\to\infty} \frac{4n*log(n+2)}{n^2*log(n)} + \frac{(n+2)^2*log(\frac{n}{2})}{n^2*log(n)}$

1.3.4. $\lim_{n\to\infty} \frac{4*log(n+2)}{n*log(n)} + 1 + \frac{4}{n} + \frac{4}{n^2} - \frac{log(2)}{log(n)} - \frac{4*log(2)}{n*log(n)} - \frac{4*log(2)}{n^2 log(n)}$

1.3.5. $= 1 + 1 + 0 + 0 + 0 + 0 + 0$

1.3.6. $= 2$

1.4. This equation evaluates to a constant result, therefore $2^{n+1} + 3^{n-1} \in \theta(3^n)$

1.4.1. $2^{n+1} + 3^{n-1} \approx \in \theta\,(3^n)$

1.4.2. $\lim_{n\to\infty} \frac{2^{n+1}+3^{n-1}}{3^n}$

1.4.3. $\lim_{n\to\infty} \frac{2^n*2}{3^n} + \frac{3^n*3^{-1}}{3^n}$

1.4.4. $\lim_{n\to\infty} \frac{2^n*2}{3^n} + \frac{1}{3}$

1.4.5. $= 0 + \frac{1}{3}$

1.4.6. $= \frac{1}{3}$

1.5. This equation evaluates to a constant result, therefore

1.5.1. $\lfloor log_2 n \rfloor \approx log_2(n)$

1.5.2. $\lim_{n\to\infty} \frac{\lfloor log_2 n \rfloor}{log_2 n}$

1.5.3. $\lim_{n\to\infty} \frac{log_2 n}{log_2 n} - \frac{\varepsilon}{log(n)}$

1.5.4. $= 1 + 0$

1.5.5. $= 1$

2. $5 * log(n + 100)^{10}$, $[\ ln(n)\ ]^2$, $\sqrt[3]{n}$, $0.001n^4 + 3n^3 + 1$, 3^n, $(n-2)!$

2.1. $(n-2)! \approx n!$

2.2. $5 * log(n + 100)^{10} \approx log(n)$

2.3. $2^{2n} \approx 2^n$

2.4. $0.001n^4 + 3n^3 + 1 \approx n^4$

2.5. $[\ ln(n)\ ]^2 \approx [\ log_e(n)\ ]^2 \approx [\ log(n)\ ]^2$

2.6. $\sqrt[3]{n} \approx n^{1/3}$

2.7. 3^n

3.

3.1.

3.1.1. $= \sum_{i=1}^{500}(2i - 1)$

3.1.2. $= \sum_{i=1}^{500} 2i - \sum_{i=1}^{500} 1$

3.1.3. $= 2\sum_{i=1}^{500} i - 500$

3.1.4. $= 2\frac{[500(500+1)]}{2} - 500$

3.1.5. $= 25000$

3.2.

3.2.1. $\sum_{i=1}^{10} 2^i$

3.2.2. $= 2\frac{1-2^{10}}{1-2}$ (using geometric sequence sum formula)

3.2.3. $= 2\frac{-1023}{-1}$

3.2.4. $= 2046$

3.3.

3.3.1. $= \sum_{i=3}^{n+1} 1$

3.3.2. $= u - l + 1$ (applying sum rule for $\sum_{i}^{n} 1$)

3.3.3. $= n + 1 - 3 + 1$

3.3.4. $= n - 1$

3.4.

3.4.1. $\sum_{i=3}^{n+1} i$

3.4.2. $= \sum_{i=1}^{n+1} i - \sum_{i=1}^{2} i$

3.4.3. $= \frac{(n+1)(n+2)}{2} - 3$

3.5.

3.5.1. $\sum_{i=0}^{n-1} i(i + 1)$

3.5.2. $= \sum_{i=0}^{n-1} i^2 + \sum_{i=0}^{n-1} i + 0$

3.5.3. $= \frac{(n-1)(n)[2(n-1)+1]}{6} + \frac{n\,(n-1)}{2}$

3.5.4. $= \frac{(n-1)(n)(2n-1)+(3)(n)(n-1)}{6}$

3.5.5. $= \frac{(n-1)(n)(2)(n+1)}{6}$

3.5.6. $= \frac{(n-1)(n)(n+1)}{3}$

3.6.

3.6.1. $\sum_{j=1}^{n} 3^{j+1}$

3.6.2. Geometric sequence $a_n = a_0 * r^{n-1}$

3.6.2.1. $a_j = 3^{j+1},\ a_{j+1} = 3^{j+1+1},\ a_0 = 3$

3.6.2.2. Finding r, $r = \frac{3^{j+1+1}}{3^{j+1}} = 3$

3.6.2.3. $a_n = 3 * 3^{n-1}$

3.6.3. Geometric sequence sum $\frac{a_1(1-r^n)}{1-r}$

3.6.3.1. $= 9\frac{1-3^n}{1-3}$

3.6.3.2. $=-\ \frac{9(1-3^n)}{2}$ (this is the solution)

3.7.

3.7.1. $\sum_{i=1}^{n}\sum_{j=1}^{n} i\,j$

3.7.2. $= \sum_{i=1}^{n} i \sum_{j=1}^{n} j$

3.7.3. $= \sum_{i=1}^{n} \frac{i(n)(n+1)}{2}$

3.7.4. $= \frac{n(n+1)}{2} \sum_{i=1}^{n} i$

3.7.5. $= \frac{n^2(n+1)^2}{4}$

3.8.

3.8.1. $\sum_{i=1}^{n} \frac{1}{i(i+1)}$

3.8.2. $= \sum_{i=1}^{n} \frac{1}{i} * \sum_{i=1}^{n} \frac{1}{i+1}$

3.8.3. $= \sum_{i=1}^{n} \frac{1}{i} * (\sum_{i=1}^{n} 1 + \sum_{i=1}^{n} \frac{1}{i})$

3.8.4. $= ln(n) + \gamma * (n + ln(n) + \gamma)$

4.

4.1.

4.1.1. $\sum_{i=0}^{n-1} (i^2 + 1)^2$

4.1.2. $= \sum_{i=0}^{n-1} i^4 + 2i^2 + 1$

4.1.3. $= \sum_{i=0}^{n-1} i^4 + \sum_{i=0}^{n-1} 2i^2 + \sum_{i=0}^{n-1} 1$

4.1.4. $\in [\ \theta\ (n^5) + \theta(n^3) + \theta(n)\ ] \in \theta\ (n^5)$

4.2.

4.2.1. $= \sum_{i=2}^{n-1} log(i^2)$

4.2.2. $= \sum_{i=2}^{n-1} 2 * log(i) \in \theta\ [\ n\ log(n)\ ]$

4.3.

4.3.1. $= \sum_{i=1}^{n} (i + 1) * 2^{i-1}$

4.3.2. $= \sum_{i=1}^{n} i * 2^{i-1} + 2^{i-1} \in \theta\ (n^2 2^n)$

4.4.

4.4.1. $= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i + j)$

4.4.2. $= \sum_{i=0}^{n-1} (i^2 + j^2) \in \theta(n^3) + \theta(n^3) \in \theta(n^3)$

5. $D(n) = 2,\ M(n) = n + 1,\ S(n) + A(n) = 2n$

6.

6.1. This algorithm computes a sorted array from greatest to least using insertion sort.

6.2. The input size is n because the input contains n real numbers

6.3. The basic operation is the key comparison of A[j] > A[i]

6.4. The amount of times the basic operation is executed can be represented by the summation shown on the left hand side of this equation. To calculate this for any value of n, we can translate this to

$$\sum_{i=0}^{n-1} n-1-i = n^2 - n - \frac{n(n-1)}{2}$$

what is shown by the right hand side to see our final answer.

6.4.1.

6.5. $X \in \Theta (n^2)$

7.

7.1. $a_n = 4 * 3^{n-1}$ **for** $n > 1$

7.1.1. The constant ratio is 3. Measured by dividing A(n) by A(n-1)

7.1.2. Geometric sequence $a_n = a_1 * r^{n-1}$

7.1.2.1. $a_n = 4 * 3^{n-1}$

7.1.3. $4 * 3^{n-1}\ \varepsilon\ \theta (3^n)$

7.2. $a_n = 5 (n-1)$ **for** $n > 1$

7.2.1. This is an arithmetic sequence with a general progression formula of $a_n = a_1 + (n - 1)d$

7.2.2. The constant difference is found by measuring the difference between adjacent terms.

7.2.2.1. $5 - 0 = 5,\ 10 - 5 = 5,\ 15 - 10 = 5,\ 20 - 10 = 5$

7.2.2.2. $d = 5$

7.2.3. Substituting into the general progression formula we get $a_n = 5 (n-1)$

7.2.4. $5 (n-1)\ \varepsilon\ \theta (n)$

7.3. $a_n = \frac{n(n+1)}{2}$ **for** $n > 0$

7.3.1. The first few terms for this sequence

7.3.1.1. $A(0) = 0, A(1) = 1, A(2) = 3, A(3) = 6$

7.3.1.2. Quickly, we are able to identify that this is a sequence of triangular numbers

7.3.1.3. Therefore, its solution is $T_n = \sum_{k=1}^{n} k$, or alternatively $a_n = \frac{n(n+1)}{2}$

7.3.1.4. $a_n = \frac{n(n+1)}{2} \in \theta (n^2)$

7.4. Picture

$A(n) = A(n/5) + 1$

$* \; A(n/5) = A(n/25) + 1$

sub. $A(n) = A(n/25) + 2$

$* \; A(n/25) = A(n/125) + 1$

sub. $A(n) = A(n/125) + 3$

To simplify...

$A(5^k) = A(5^k/5) + 1$

$A(5^k) = A(5^{k-1}) + 1$

or...

$A(5^k) = A(5^{k-i}) + i = A(5^{k-k}) + k$

$= A(5^0) + k = A(1) + k = 1 + k$

To put in terms of n, use $n = 5^k$

$n = 5^k$

$k = \log_5 n$

so... $A(5^k) = \log_5 n + 1 \in \Theta \log(n)$

7.5. Picture

$$A(n) = 2A(n/2) + n - 1 \qquad n = 2^k$$

$$A(2^k) = 2A(2^{k-1}) + 2^k - 1 \qquad \log(n) = k\log(2)$$

$$k = \frac{\log(n)}{\log(2)}$$

$$* \; A(2^{k-1}) = 2A(2^{k-2}) + 2^{k-1} - 1$$

sub.

$$A(2^k) = 2^2 A(2^{k-2}) + 2^{k+1} - 3$$

$$A(2^k) = 2^i A(2^{k-i}) + 2^{k+i-1} - (2^i - 1)$$

$$= 2^k A(2^{k-k}) + 2^{k+k-1} - (2^k - 1)$$

$$= 2^k A(1) + 2^{2k-1} - 2^k + 1$$

$$= 2^k \cdot 0 + 2^{2k-1} - 2^k + 1$$

$$= 2^{2k-1} - 2^k + 1$$

$$= 2^{2k} \cdot 2^{-1} - 2^k + 1$$

$$= \frac{2^{2k}}{2} - 2^k + 1$$

$$= \frac{n^2}{2} - n + 1$$

$$A(2^k) = \frac{n^2}{2} - n + 1 \in \Theta(n^2)$$

8.

8.1. This algorithm computes the sum of all of the squares in the range 1 to n. To demonstrate this with an equation $\sum_{i=1}^{n} i^2$

8.2. The input size of this algorithm is a single integer. This integer must be greater than or equal to 1.

8.3. The basic operation of this algorithm is multiplication (n * n)

8.4. The number of times this base operation is run can be represented by

8.4.1. $B(n) = B(n-1) + 1$ for all $n > 1$

8.4.2. $B(1) = 1$

8.5. This algorithm can be represented as

8.5.1. $= \sum_{i=1}^{n} i^2$

8.5.2. $= \frac{n(n+1)(2n+1)}{6} \in \theta(n^3)$

9.

9.1. The recurrence relation representing the function's values is as follows

9.1.1. $Q(1) = 1$

9.1.2. $Q(n) = Q(n-1) + 2n - 1$ for all $n > 1$

9.1.3. Solved: $Q(n) = n^2$

9.1.4. This algorithm computes n^2

9.2. The recurrence relation representing the number of multiplications is as follows

9.2.1. $M(n) = M(n-1) + 1$ for all $n > 1$

9.2.2. $M(1) = 0$

9.2.3. Solved: $M(n) = n - 1$

9.3. The recurrence relation representing the number of additions is as follows

9.3.1. $A(n) = A(n-1) + 2$ for all $n > 1$

9.3.2. $A(1) = 0$

9.3.3. Solved: $A(n) = 2(n-1)$

10.

10.1. This algorithm completes a binary search on the given array and returns the index of K if the integer is present and -1 otherwise

10.2. The input size relative to the other parameters is expressed as such

  10.2.1. l can be any integer such that $l \geq 0$

  10.2.2. r can be any integer such that $r \geq 0$

  10.2.3. A is an array of $r - l$ integers

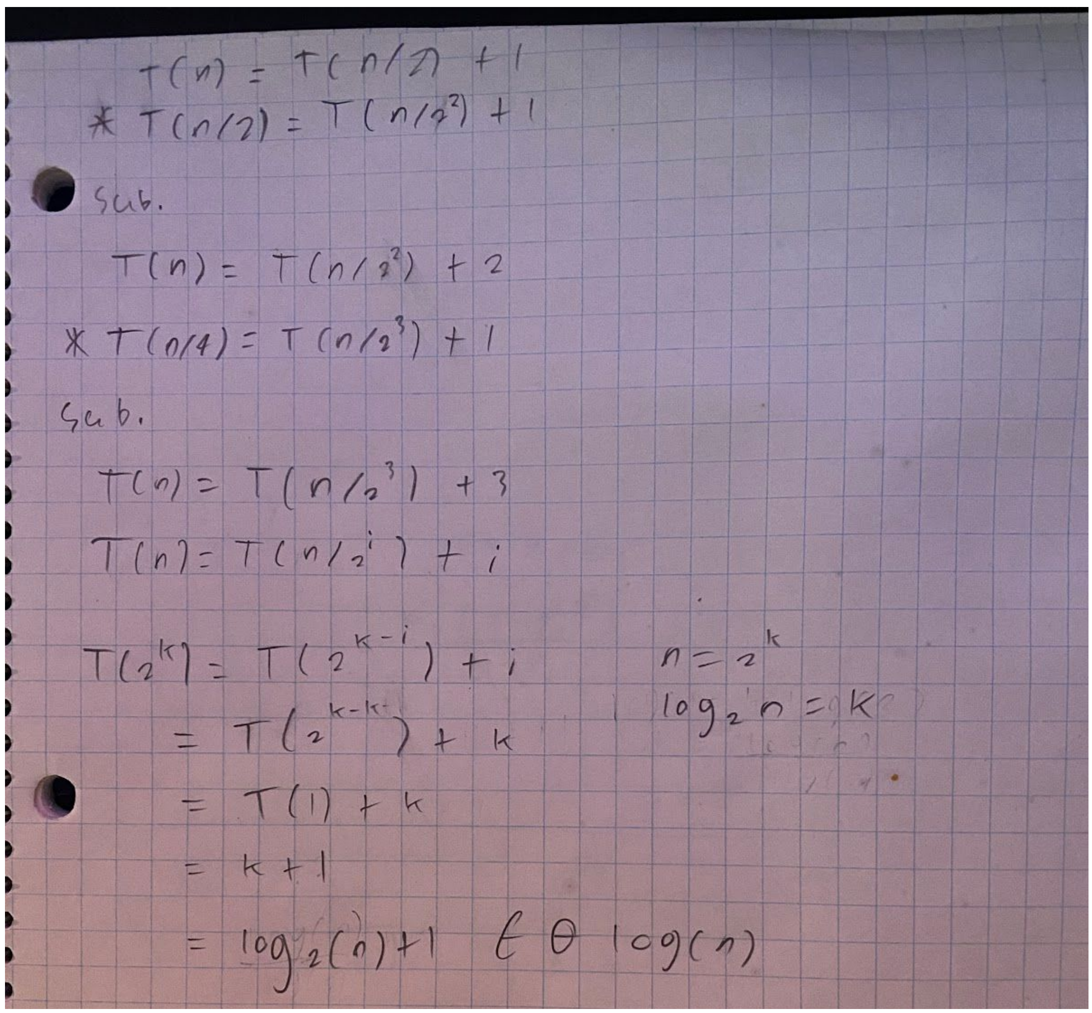
  10.2.4. K can be any integer

10.3.

  10.3.1. $T(1) = 1$

  10.3.2. $T(n) = T(n/2) + 1$

  10.3.3. Solved



$T(n) = T(n/2) + 1$

$* \; T(n/2) = T(n/2^2) + 1$

Sub.

$T(n) = T(n/2^2) + 2$

$* \; T(n/4) = T(n/2^3) + 1$

Sub.

$T(n) = T(n/2^3) + 3$

$T(n) = T(n/2^i) + i$

$T(2^k) = T(2^{k-i}) + i$ $\qquad n = 2^k$

$= T(2^{k-k}) + k$ $\qquad \log_2 n = k$

$= T(1) + k$

$= k + 1$

$= \log_2(n) + 1 \in \Theta \log(n)$

10.4. It will have the same efficiency class because of the smoothing rule.