Colin Smith and Emma Steinman
CS 271 Project 5
December 16, 2017

1.a By definition $T(n) = T(k) + T(n-k-1) + cn : n \geq 2$ and $T(n) = a : n \leq 1$

Our hypothesis is $T(n) = (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a$. The base case occurs when $n \leq 1$.

$RHS : (\frac{c}{2(k+1)})0^2 + (\frac{T(k)}{k+1} + \frac{c}{2})0 + a = 0 + 0 + a = a$

$LHS : T(0) = a$

$RHS = LHS$

Assume $T(n) = (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a$ up to $n-1$. Now in the inductive step we prove that $T(n) = T(k) + T(n-k-1) + cn = (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a$.

Due to the inductive assumption because $k \in \mathbb{N} \Leftrightarrow n-k-1 \leq n-1$:

$T(n) = T(k) + T(n-k-1) + cn = T(k) + (\frac{c}{2(k+1)})(n-k-1)^2 + (\frac{T(k)}{k+1} + \frac{c}{2})(n-k-1) + a + cn$
$= T(k) + (\frac{c}{2(k+1)})n^2 - (\frac{c}{2(k+1)})2nk - (\frac{c}{2(k+1)})2n + (\frac{c}{2(k+1)})k^2 + (\frac{c}{2(k+1)})2k + (\frac{c}{2(k+1)}) + (\frac{T(k)}{k+1} + \frac{c}{2})n - (\frac{T(k)}{k+1} + \frac{c}{2})k - (\frac{T(k)}{k+1} + \frac{c}{2}) + a + cn$
$= [(\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a] + T(k) - (\frac{c}{2(k+1)})2nk - (\frac{c}{2(k+1)})2n + (\frac{c}{2(k+1)})k^2 + (\frac{c}{2(k+1)})2k + (\frac{c}{2(k+1)}) - (\frac{T(k)}{k+1} + \frac{c}{2})k - (\frac{T(k)}{k+1} + \frac{c}{2}) + cn$

$Note : [(\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a]$ will be called S until the end of the inductive step to keep things concise

$S + T(k) - \frac{cnk}{k+1} - \frac{cn}{k+1} + \frac{ck^2}{2(k+1)} + \frac{2ck}{2(k+1)} + \frac{c}{2(k+1)} - \frac{T(k)k}{k+1} - \frac{c}{2} + cn$

$S + T(k) - \frac{cn(k+1)}{k+1} + \frac{c(k^2+2k+1)}{2(k+1)} - \frac{T(k)(k+1)}{k+1} - \frac{c(k+1)}{2} + cn$

$S + T(k) - cn + \frac{c(k+1)^2}{2(k+1)} - T(k) - \frac{c(k+1)^2}{2(k+1)} + cn$

S

$S = (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a$

Thus we have proven our n case as we have shown $T(n) = (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a$

1.b $\Theta(n^2) = 0 \leq c_1 n^2 \leq (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a \leq c_2 n^2$
$= \frac{1}{3(k+1)}n^2 \leq (\frac{c}{2(k+1)})n^2 + (\frac{T(k)}{k+1} + \frac{c}{2})n + a \leq 2(c+a)n^2$ $\frac{1}{3}n^2 \leq (\frac{c}{2}n^2 + \frac{T(k)}{k+1} + \frac{c}{2})n + a \leq 2(c+a)n^2$

For $n = 1$

$\frac{1}{3(k+1)} \leq \frac{c}{2(k+1)} + \frac{T(k)}{k+1} + \frac{c}{2} + a \leq 2(c+a)$

Since k can be at least 0 and at most $n-1$, prove for both.

for $k = 0$

$\frac{1}{3} \leq \frac{c+2T(0)+c(1)}{2(0+1)} \leq 2(c+a)$

$\frac{1}{3} \leq \frac{c+2a+c}{2} \leq 2(c+a)$

$\frac{1}{3} \leq c + a \leq 2(c+a)$ Which is true for all c

For $k = n-1$

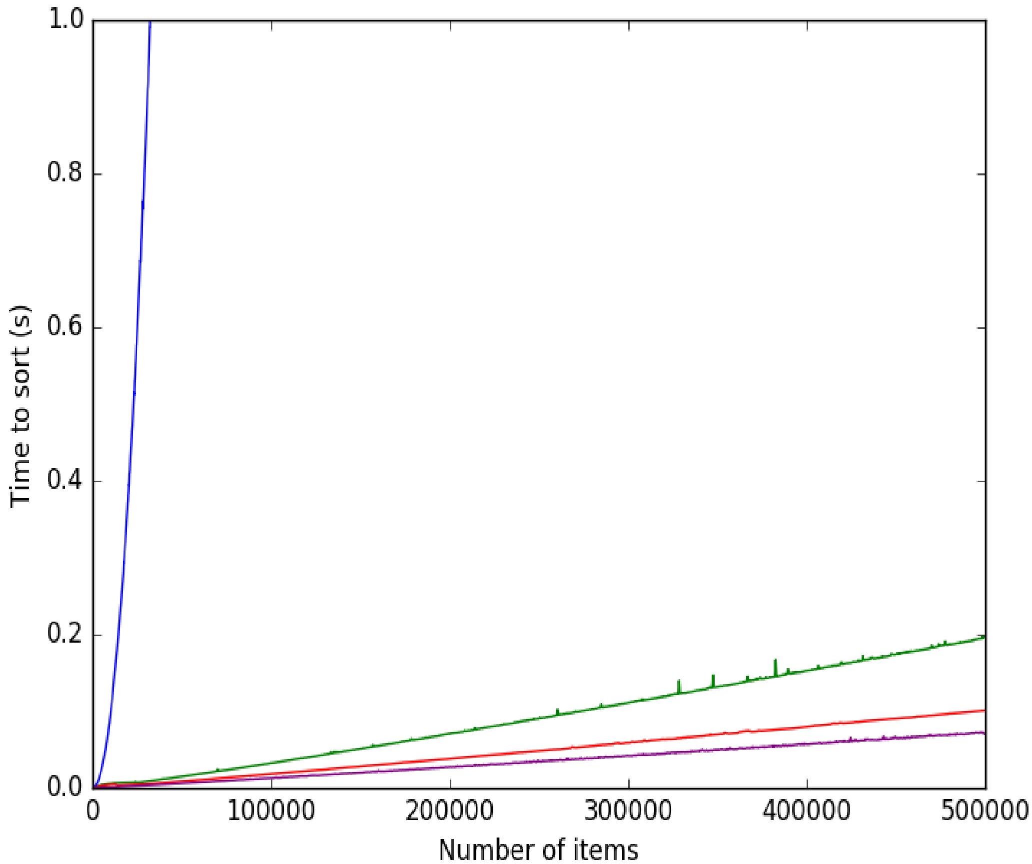
$\frac{1}{3(n-1)+1}n^2 \leq \frac{c}{2((n-1)+1)}n^2 + (\frac{T(n-1)}{(n-1)+1} + \frac{c}{2})n + a \leq 2(c+a)n^2$

$\frac{n^2}{3n} \leq \frac{cn}{2} + (\frac{T(n-1)}{n} + \frac{c}{2})n + a \leq 2cn^2 + 2an^2$

$\frac{n}{3} \leq \frac{cn}{2} + T(n-1) + \frac{cn}{2} + a \leq 2cn^2 + 2an^2$

$\frac{n}{3} \leq cn + T(n-1) + a \leq 2cn^2 + 2an^2$ Which is true for any c and n

1. Graph of Sort Times



Blue = Insertion Sort
Green = Heap Sort
Red = Merge Sort
Purple = Quick Sort

In this plot you can see that quick sort is the fastest sorting algorithm, followed by merge sort, followed by heap sort, with insertion sort being the slowest. We tested all of our sorting algorithms on randomized arrays ranging from size 0 to 500,000, by increments of 500 for Heap Sort, Merge Sort, and Quick Sort and from 0 to 50000 in increments of 100 for Insertion Sort. This plot follows the time complexities we found for each of these sorting algorithms. Insertion sort has time complexity $\Theta(n^2)$ which is much slower than the next fastest sorting algorithm, heap sort. Heap sort, merge sort, and quick sort all have time complexity of $\Theta(nlogn)$. However, because Θ is purely asymptotic it does not take into account constants which make certain algorithms faster or slower even though they are asymptotically similar; therefore, though Heap Sort, Merge Sort, and Quick Sort are all $\Theta(nlogn)$, due to constants Quick Sort is the quickest.