Zijian Gao
26319560
2/29/2016
CS311 HW2

4.2-3.)

Strassen's method can be modified to operate on n x n matrixes where n is not an exact power of two by right zero padding the input matrixes A and B such that they are m x m matrixes where m is the ceiling of log(n). In other words n < m where $m = 2^i$ such that $2^{i-1} < n < 2^i$, then $m = 2^{\lceil \log(n) \rceil}$. Pad matricies A and B such that all n x n original values are up-left justified, in other words they exist in indexes [0 to n-1] x [0 to n-1], and padded zeroes are in indexes [n to m - 1] x [n to m - 1]. Then, the sums for the original matrix A and B will be similarly top-left justified and can simply be extracted from the resulting matrix.

To find the change in runtime, first find an upper limit on m relative to n. As m is defined $2^{i-1} < n < 2^i$, use $2^{i-1} < n$. $2^i < 2n$ follows by multiplication, or $m < 2n$. Since Strassen's Method is given to take $\theta(n^{\log 7})$ time, it follows that the modified version will take $\theta((2n)^{\log 7}) = \theta(2^{\log 7} \bullet n^{\log 7})$ which is proof that the change in the upper bound for the function is a constant factor, which can be disregarded in the analysis as part of the constant multiple and so this method would still take $\theta(n^{\log 7})$ time.

4.2-4.)

Modifying again the runtime of Strassen's Method, $T(n) = 7T(n/2) + \theta(n^2)$ becomes $T(n) = kT(n/3) + \theta(n^2)$. This is because the base case of Strassen's Method is reached at a 2x2 matrix, by recursing on a quarter of the input matrix, so for the base case of a 3x3 matrix, recurse on a ninth of the input matrix. Since the master method gives $\theta(n^{\log 7})$ runtime for Strassen's, use that to establish an upper bound on the 3x3 version. $\theta(n^{\log_3 k}) \le \theta(n^{\log 7})$ as this is looking for k such that the runtime is still bounded by Strassen's runtime. Taking the n'th log of both sides, $\log_3 k \le \log 7$ and so $k \le 3^{\log 7}$, so finally $k \le 21.85$ and the maximum value for k is the floor of that, 21, because k must be a whole number and 21.85 is the upper bound.

For runtime, substitute k into the previous equation giving $T(n) = 21T(n/3) + \theta(n^2)$ which is $\theta(n^{\log_3 21}) \approx \theta(n^{2.771})$ by the master method a > b^k.