1. Administrivia: see course webpage http://www.cse.wustl.edu/~bjuba/cse347/s19

2. Matrix Multiplication via Divide and Conquer
   (a) Matrix multiplication: for $n \times n$ matrices $A, B$, $C = AB$ is the $n \times n$ matrix where $C_{ij} = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$
   (b) Naive algorithm is $O(n^3)$
   (c) What if we write $A = \begin{bmatrix} A_{1,1}^{(1,1)} & A_{1,2}^{(1,2)} \\ A_{2,1}^{(2,1)} & A_{2,2}^{(2,2)} \end{bmatrix}$, $B = \begin{bmatrix} B_{1,1}^{(1,1)} & B_{1,2}^{(1,2)} \\ B_{2,1}^{(2,1)} & B_{2,2}^{(2,2)} \end{bmatrix}$, $C = \begin{bmatrix} C_{1,1}^{(1,1)} & C_{1,2}^{(1,2)} \\ C_{2,1}^{(2,1)} & C_{2,2}^{(2,2)} \end{bmatrix}$?
   Then $C_{1,1}^{(1,1)} = A_{1,1}^{(1,1)}B_{1,1}^{(1,1)} + A_{1,2}^{(1,2)}B_{2,1}^{(2,1)}$, etc.
   (d) Recursive algorithm has 8 subproblems of size $n/2$, gives $n^3$ time again

3. Strassen’s Algorithm
   (a) If we can use seven subproblems instead of eight, we get a running time of $n^{2.81}$
   (b) Strassen’s algorithm achieves this with the following subproblems
      \[ P_1 = A_{1,1}^{(1,1)}(B_{1,2}^{(1,2)} - B_{2,2}^{(2,2)}) \]
      \[ P_2 = (A_{1,1}^{(1,1)} + A_{1,2}^{(1,2)})B_{2,2}^{(2,2)} \]
      \[ P_3 = (A_{2,1}^{(2,1)} + A_{2,2}^{(2,2)})B_{1,1}^{(1,1)} \]
      \[ P_4 = A_{1,2}^{(1,2)}(A_{2,1}^{(2,1)} - B_{1,1}^{(1,1)}) \]
      \[ P_5 = (A_{1,1}^{(1,1)} + A_{2,2}^{(2,2)})(B_{1,1}^{(1,1)} + B_{2,2}^{(2,2)}) \]
      \[ P_6 = (A_{1,2}^{(1,2)} - A_{2,2}^{(2,2)})(B_{2,1}^{(2,1)} + B_{2,2}^{(2,2)}) \]
      \[ P_7 = (A_{1,1}^{(1,1)} - A_{2,1}^{(2,1)})(B_{1,1}^{(1,1)} + B_{1,2}^{(1,2)}) \]
   combined the following way
      \[ C_{1,1}^{(1,1)} = P_5 + P_4 - P_2 + P_6 \]
      \[ C_{1,2}^{(1,2)} = P_1 + P_2 \]
      \[ C_{2,1}^{(2,1)} = P_3 + P_4 \]
      \[ C_{2,2}^{(2,2)} = P_5 + P_1 - P_3 - P_7 \]