Joseph Turnbull HW06 CIS 677

1. 99% as even with a perfect efficiency we still have that 1% of code that must run sequentially

2. Algorithm A is cost-optimal as it's cost is N processors * the time it took using those processors log (n) so cost was n log n which is asymptotically equivalent to the sequential speed which was n log n.

Algorithm B is not cost-optimal. Cost = n^2*n = n^3 which is not asymptotically equivalent to the sequential speed which was n log n.

3. Taking AMDAHL's law as a limit we can see that maximum serial code percentage we can have is 10%

4. By using Gustafson's Law with a serial percentage of 10/240 we have a speedup of 8+(1-8)(10/240) = 7.7083

5a) It does not scale well as by keeping the data set constant while we have our static time for the serial portion of the code by increasing processors we have them getting in each other's way more often

b) It scales better as although we are increasing our read time of the data, we have greater partialization as the data set is much larger giving the given processors good room to attack the problem.