Problem Set #3
IE 495
Due September 25

Written Problems

1. A binary relation $\equiv$ on a set A is said to be an equivalence relation on A if it is reflexive, symmetric, and transitive, i.e. for all $a, b, c \in A$,

   $a \equiv a$,
   $a \equiv b \Rightarrow b \equiv a$, and
   $(a \equiv b) \wedge (b \equiv c) \Rightarrow a \equiv c$.

   Furthermore, a *partial order* on a set A is defined by a relation "$\leq$" that is reflexive, transitive, and also obeys the property that for $a, b \in A$

   $(a \leq b) \wedge (b \leq a) \Rightarrow b \equiv a$

   A partial order is called a *total order* if every pair of elements of A are related. Recall the set−valued functions Θ, Ω, O, o, and ω we discussed in class (see Lecture 3, slide 16). If $\mathcal{P}$ is the set of all polynomials, we will denote by $\Theta_{|\mathcal{P}}: \mathcal{P} \to 2^{\mathcal{P}}$ the function Θ restricted to just the set of polynomials and similarly for the other functions we've discussed.

   a. Use the set−valued function $\Theta_{|\mathcal{P}}$ to define an equivalence relation on $\mathcal{P}$. You must use the definition to prove your assertion.

   b. Use the set−valued function $\Omega_{|\mathcal{P}}$ to define a total order on $\mathcal{P}$.

   c. Explain how this relates to asymptotic analysis of algorithms.

2. Compare Amdahl's Law to the bounds presented in the paper by Gustafson. What assumptions does each author make? Which view do you think is more realistic? Give a concrete numerical example of the difference in predicted speedup between the two models.

3. Consider a parallel version of the merge sort algorithm discussed in class. Calculate the theoretical speedup of such an algorithm. How does this result fit into the frameworks discussed in Problem 2?

4. Is super−linear speedup theoretically possible? Is it possible in practice? Explain a situation in which you might observe super−linear speedup in practice.

5. Solve 2 of the recurrence relations in Miller and Boxer page 62.

Programming Problems

6. TBD