CS3933 Assignment 11
Summer 2007

Due: Thursday, August 2, in class or by 10:00am in the box on E-level. Please try the questions before Tuesday, and ask questions then if you want to.

- Read chapter 8 again.
- In chapter 6, question 17, page 328.
- Consider the **SET PACKING** (SP) problem. An instance is simply a finite family S of finite sets and a positive integer k. The question is whether there is a subfamily of F, consisting of k sets, such that no two of these sets contain a common element. (I am using the term *family* to mean a set of sets, to avoid confusion I hope.)

  Prove that Set Packing is in NP.
- Prove that **Independent Set** polynomially reduces to **Set Packing**, ie. $IS \leq_P SP$. This completes a proof that Set Packing is NP-complete.
- In chapter 8, do question 1 on page 505.

  To solve a dynamic programming problem, you must define a mathematical function *in terms of the inputs*, then give a recurrence relation that it must satisfy (including a base case), and explain how you can calculate the function iteratively.

  To prove a problem P is NP-complete, you must

  1. Show that the problem P is in NP.
  2. Choose some other known NP-complete problem Q.
  3. Reduce Q to P, ie. show that $Q \leq_P P$.