Instructions for the second project

The instructions do not change much from the first project. The main difference is that for this one I really want you to focus more on applications and less on theory, if possible.

- All projects are to be done in groups, unless you are really strongly opposed to group work, or have a schedule that doesn't work with other people's schedules. You may work with a partner or in a group of three (possibly four). You may change groups for the different projects.
- Project 2 consist of picking a section from the book, reading the section, picking one or two main ideas, finding additional resources if possible (some are provided), writing up a summary of the main ideas, and giving a 5-7 minute presentation in class. Each group must choose a different topic.
- The summary of the section needs to be typed, and should be 2-5 pages long (you can write symbols by hand if necessary). The summary will be due on May 7. Summaries are worth 15 points.
- For the presentation, pick the most interesting part of your summary, especially if you looked at a topic that has applications in computer science or real life. Summaries are worth 10 points. Make sure that all group members participate in the presentation. You will give your presentation on May 5.
- Be creative. Feel free to use outside sources and add information that you think is relevant and interesting.

Topics

- The first topic is not book related. You can
  - Submit a program/code you have used before that uses logic, counting techniques, Big-O notation, induction, or any other topic we have encountered or will encounter in this course. If you are not sure if a topic qualifies, ask.
  - Write a short paper or summary of a mathematical program you had to solve in programming. The math concepts need to be related to course material.

  - Write a short paper or summary of how the concepts from our class can be used in computer science.

- Section 5.6. Only look at the Pigeonhole principle. Do not worry about the proofs in the section that depend on it, but look at problems 28 and 29 and try to solve them. Then look at this website:

  http://mindyourdecisions.com/blog/2008/11/25/16-fun-applications-of-the-pigeonhole-principle/

  and choose some examples from it to present.

- Section 6.5: Applications of Combinatorics. This section is interesting in itself, and you can definitely choose to present some problems from it, but I would be interested in hearing more about gambling. In particular, I found a site called "Casino Games Online," that has the following interesting topics:

  http://www.casino-games-online.biz/dice/chevalier-de-mere.html – the problem that started probability theory in the 16th century

  http://www.casino-games-online.biz/probability/probability-odds-luck.html – some definitions in the context of gambling

  http://www.casino-games-online.biz/probability/probability-assessment.html – common myths

  http://www.casino-games-online.biz/probability/true-odds-house-edge.html – true odds

- Section 7.1: Inclusion-exclusion principle. Focus only on derangements. Here is more information on the topic:

  http://www.wmueller.com/precalculus/e/e6.html

- Section 7.2: Generating functions. This is a very commonly used tool in mathematics. Look at this page for some examples (in particular, look at the "Rolling the dice simultaneously" example).

  http://www.wmueller.com/precalculus/e/e6.html

- Section 7.5: Combinatorial proofs. This section is not really applied, but is interesting because you use combinatorial arguments to show numerical results instead of the other way around. The book gives good examples, but here is another resource:

  http://www.cut-the-knot.org/arithmetic/combinatorics/CombinatorialProofs.shtml

- Section 8.5. You can only do Markov Chains and skip the first part of the section. Below are some interesting examples, but the book has some too.

  http://en.wikipedia.org/wiki/Examples_of_Markov_chains

  http://www.sosmath.com/matrix/markov/markov.html

  However, the best application of all that I have found is the one for computing probabilities for landing on each space in Monopoly:

  http://www.tkcs-collins.com/truman/monopoly/monopoly.shtml

  Now that's a fun project!