Math 350 - Combinatorial Algorithms

Spring 2009 - Lehigh University (Contact Garth Isaak, gisaak@lehigh.edu for more information.)

This is a topics course in discrete mathematics. Roughly, the topic is algorithms for generating certain combinatorial objects (partitions, permutations, combinations, trees etc) and the mathematical theory behind these algorithms. The course will include student projects and presentations.

Over thirty years ago Donald Knuth wrote a three volume set titled 'The Art of Computer Programming' which has become a classic reference. Large portions of the six chapters in these volumes consist of mathematics exercises and sketches of their solutions. Knuth is currently working on a 7th chapter, which is planned to consist of three volumes. Recently, 4 fascicles (each a 100+ page portion of the 7th chapter) have been published titled: Introduction to combinatorial algorithms and boolean functions; Generating all tuples and permutations; Generating all combinations and partitions; Generating all trees & History of combinatorial generation. There is far more material than can be covered in a semester. We will be looking at portions of these fascicles (and possibly reviewing some more basic elements of combinatorics). Students will be expected to develop the proof sketches for exercises into notes and presentation for the class as well as work on a project or projects exploring more substantial problems.

There are no formal prerequisites. However, students should have taken at least one mathematically oriented class where they have encountered writing proofs.

For illustration here are some problems that give a hint of some of the sorts of problems that we can examine. These are related to old results on what are called Gray codes and Debruijn cycles (both also happen to have real applications).

1. Consider $\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{matrix}$. This is a list of all (eight) binary 3-tuples in order so that each item differs from the previous by one bit and the same is true for the first and last item. This is equivalent to $\emptyset, \{3\}, \{2,3\}, \{2\}, \{1,2\}, \{1,2,3\}, \{1,3\}, \{1\}$, a circular list of the eight subsets of $\{1,2,3\}$ so that adjacent sets differ by exactly one element. It is also equivalent to a traveling salesperson tour through a cube, visiting each vertex exactly once, following edges and returning home.

1a. Create examples like above for binary 4-tuples and 5-tuples.

1b. Describe an algorithm to create the same list as 1a for n-tuples and prove that it is correct.

1c. Describe an algorithm as in 1b so that one can determine (without generating the whole list) the k^{th} item in the list.

1d. Determine whether or not it is possible to list binary $2k+1$-tuples with exactly k or $k+1$ ones in the same manner as above, adjacent items on the list differ in exactly one bit. (This is an unsolved problem.)

2. Consider 00011101 and 0000111101100101. Write these strings on a circle. Each of the eight possible binary 3-tuples appears exactly once consecutively in the first example and each of the 16 possible binary 4-tuples appears exactly once consecutively in the second.
2a. Create examples like above for binary 5-tuples and for ternary 3-tuples.
2b. Prove that the prefer-ones algorithms works for general n: write down n zeroes then for each further bit write a one unless doing so repeats an n-tuple that has already appeared.
2c. Describe an algorithm as in 2b so that one can determine without generating the whole list the k^{th} bit in the list.
2d. Prove that there are essentially $2^{2^{n-1}}/2^n$ different such listings for n-tuples and generalize to k-ary n-tuples (where k symbols are used instead of 2).
2e. Determine whether or not it is always possible to construct a $2^{mn} \times 2^{mn}$ binary array that when it is wrapped on a torus contains each $m \times n$ binary array exactly once. (If we replace binary with k-ary it is possible if k is a prime power and an unsolved problem when k is not a prime power.)