CSI 5165 Combinatorial Algorithms | Fall 2013
Computer Science | University of Ottawa

**Homework Assignment #2** (100 points, **weight 20%** - note weight change as the last assignment will be shorter worth only 10%)
Due: November 6 (in lecture)

---

1. (35 points) **Steiner Triple Systems by backtracking**

   A Steiner Triple System of order $n$, STS($n$), is pair $(X, \mathcal{B})$ where $X$ is an $n$-set and $\mathcal{B}$ is a colection of $n(n-1)/6$ 3-subsets (triples) of $X$, and every pair of elements of $X$ is contained in exactly one triple.

   Examples: of STS(7) and STS(9), respectively:
   $([1,7], \{\{1,2,3\},\{1,4,5\},\{1,6,7\},\{2,4,6\},\{2,5,7\},\{3,4,7\},\{3,5,6\}\}$
   $([1,9], \{\{1,2,3\},\{1,4,7\},\{1,5,9\},\{1,6,8\},\{2,4,9\},\{2,5,8\},\{2,6,7\},\{3,4,8\},\{3,5,7\},\{3,6,9\},\{4,5,6\},\{7,8,9\}\})$

   (a) Write a backtracking algorithm (pseudocode) to find all STS($n$).

   (b) Implement your algorithm and use it to find the number of different STS(7) and STS(9). Can you also find the number of different STS(13) or STS(15)? Remember that an STS($n$) exists if and only if $n \equiv 1, 3 \pmod 6$. For each run of your algorithm, specify $n$, number of nodes in the backtracking tree, total time spent by the algorithm and number of STS($n$) found. Also, please show some of the STS's generated for verification.

   Hint: this can be modelled as a maximum clique problem or as an exact cover problem, so many enhancements on a basic backtracking are possible which should allow you to run the method for higher values of $n$.

   It may be useful to use the algorithm for generating 3-sets in lexicographical order (rank, unrank, successor). This is implemented in C, and available from the textbook's web page:
   `http://www.math.mtu.edu/~kreher/cages/Src.html`

2. (30 points) **Steiner Triple Systems estimation**

   Consider the algorithm you designed in question 1 to generate all Steiner triple systems of order $n$.

   - Use Knuth's method to estimate the backtracking tree size for the cases of $n = 7, 9, 13, 15$. Please, for each $n$, provide some analysis by varying the number of probes used, and showing the various estimations obtained. Note that for $n = 7, 9$ you are able to compare the estimation with the actual tree size you measure in last assignment (please include the actual tree size).
   - In this part, you will use a more general version of the theorem seen in class (Knuth 1975): associate with each node $X$ in the backtracking tree, a weight $w(X)$. The goal is to estimate $w(T) = \sum_{X \in T} w(X)$. The estimate $W$ obtained from the path $X_0, X_1, \ldots, X_k$, following the same notation as in the class notes, is:

     $$W = w(X_0) + |\mathcal{C}_0| w(X_1) + |\mathcal{C}_0||\mathcal{C}_1| w(X_2) + \cdots + |\mathcal{C}_0||\mathcal{C}_1| \cdots |\mathcal{C}_{k-1}| w(X_k).$$

     The revised theorem states that the expected value of $W$ returned by the algorithm is $w(T)$. The previously seen theorem is equivalent to $w(X) = 1$ for all nodes $X$ in the backtracking tree $T$.

     Use a method based on this generalized theorem to estimate the **number** of Steiner triple systems of order $n$; indeed you can use a weight function that assigns weight 1 to the leaves that correspond to STSs, and 0 to non-leaves and to the leaves that do not correspond to STSs. Do a similar analysis as the previous case and compare it with the actual values, which are known for $n = 7, 9, 13, 15$. The following web page the number of such systems for $n = 1, 3, 7, 9, 13, 15, 19$:

     `http://www.research.att.com/~njas/sequences/A001201`

3. (35 points) **Hill Climbing to Find Transversal Triple Systems**

A transversal triple system TTS($n$) is a set system $(X, \mathcal{B})$, such that:

(a) $X = X_1 \cup X_2 \cup X_3$,with $|X| = 3n$ and $|X_i| = n$, for $i = 1, 2, 3$.

(b) For each $B \in \mathcal{B}$ we have that $|B| = 3$ and $|B \cap X_i| = 1$.

(c) For every $x \in X_i$ and $y \in X_j$ with $i \neq j$, there exists a unique block $B \in \mathcal{B}$ such that $\{x, y\} \subseteq B$.

Develop a hill-climbing algorithm to construct a transversal triple system TTS($n$). (Hint: design a heuristic similar to that used in the algorithm for constructing Steiner Triple Systems). Show your pseudo-code and an implementation program. Test your algorithm for various values of $n$ (repeat 5 times for each $n$). Report on the total number of objects visited as well on the total running time.