Released Oct 27 Due: Upload to canvas by 9:00pm on Tuesday December 3

1. Let $\mathcal{T}$ be the class of plane trees where each node has a number of children which is divisible by three. Use Lagrange Inversion to find a formula for the number T_n of such trees with n nodes.

2. Recall that the class of positive integers $\mathcal{I} = \{1, 2, 3, \dots\}$ (where the size of an integer n equals n) has generating function $I(z) = z+z^2+z^3+\dots$. In Lecture 6 we used this fact to find the generating function for the class $\mathcal{C}$ of i integer compositions of n. Here is a variation, where we want to count only those compositions which have exactly k summands, and where each summand is a positive integer of size at most r.

   Show that the number, $C_n^{(r,k)}$, of compositions of the integer n with exactly k summands, each of which is at most r is given by

   $$C_n^{(r,k)} = [z^n] \left(z \frac{1-z^r}{1-z} \right)^k$$

   For example when $r = 4$, $k = 3$ and $n = 9$, we have that $C_9^{(4,3)} = 10$ since there are 10 compositions of 9 in the following set.

   $$\{1+4+4, 2+3+4, 2+4+3, 3+2+4, 3+3+3, 3+4+2, 4+1+4, 4+2+3, 4+3+2, 4+4+1\}$$

3. (a) Write an algorithm UNRANKKSUBSET2 which inputs n, k, and r, and which outputs the k-subset $L = \{\ell_1, \ell_2, \dots, \ell_k\}$, (with $\ell_1 < \ell_2 < \cdots < \ell_k$), which has rank r in the lexicographic ordering of k-subsets of $\{1, 2, \dots, n\}$. Your algorithm should be based on the improved ranking algorithm RANKKSUBSET2 of Lecture 8, Section 2.6.

   (b) Use the procedure UNRANKSUBSET2 that you found in part (a) to find the subset UNRANKSUBSET2(n, k, r) where $n = 10$, $k = 4$ and $r = 162$. You may do this with a computer or by hand.

4. Use Prüfer sequences to help you answer the following question. How many labeled trees have n vertices, where every vertex has degree 1 or 3?

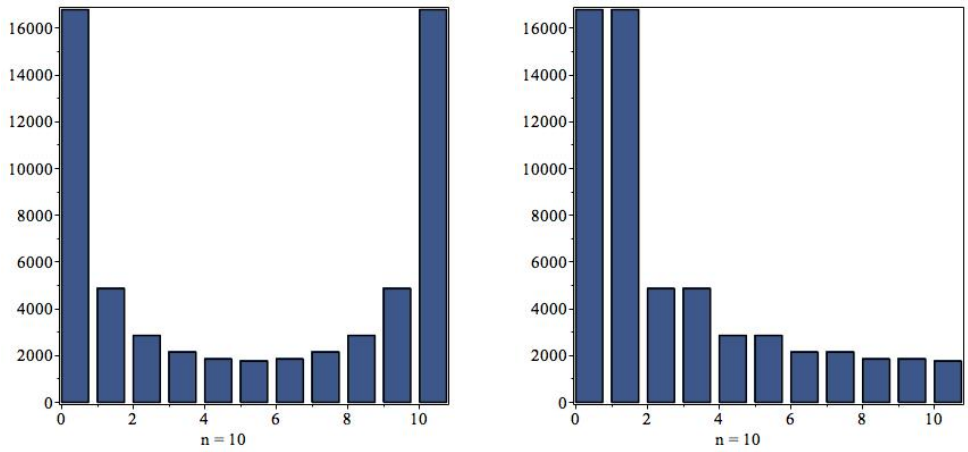
5. Find a minimal change ordering for all the Dyck paths of length $2n$, for $n = 1, 2, 3$ and 4. Note: You will first have to find a reasonable definition of "minimal change" that applies to Dyck paths. Also, you might not be able to find a *cyclic* minimal change order; the first and last Dyck paths might need to be more different than other consecutive paths.

6. You will recall that we can make a procedure which generates a random binary tree in $\mathcal{B}_n$, by taking advantage of the recursive regular specification $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$. This involves constructing a random object generator for the product class $\mathcal{B} \times \mathcal{B}$ from a random object generator for $\mathcal{B}$. We learned more generally, in Lecture 11, how to construct a random object generator for the product class $\mathcal{A} = \mathcal{B} \times \mathcal{C}$ from a random object generators for $\mathcal{B}$ and $\mathcal{C}$. To generate a random object in $\mathcal{A}_n$ we need to compare a random integer x to the sequence of numbers

   $$\begin{aligned} &B_0C_n \\ &B_0C_n + B_1C_{n-1} \\ &B_0C_n + B_1C_{n-1} + B_2C_{n-2} \\ &\vdots \\ &B_0C_n + B_1C_{n-1} + \cdots + B_nC_0, \end{aligned} \tag{1}$$

   to find the smallest integer k such that $B_0C_n + B_1C_{n-1} + \cdots + B_kC_{n-k} \geq x$.

   We learned that for binary trees we have $B_{2k+1} = \frac{1}{k+1}\binom{2k}{k}$, the kth Catalan number. We saw, for example in the Maple document `MapleRandomGeneration.mw`, that the plot of B_kB_{n-k} versus k is concave up, such as in the first plot below.



We reasoned that the random generator for $\mathcal{B} \times \mathcal{B}$ would be faster if instead we used the *boustrophedon* order

$$\begin{aligned} &B_0 B_n \\ &B_0 B_n + B_n B_0 \\ &B_0 B_n + B_n B_0 + B_1 B_{n-1} \\ &B_0 B_n + B_n B_0 + B_1 B_{n-1} + B_{n-1} B_1 \\ &B_0 B_n + B_n B_0 + B_1 B_{n-1} + B_{n-1} B_1 + B_2 B_{n-2} \\ &\vdots \end{aligned} \tag{2}$$

when seeking the appropriate value of k, since the corresponding plot for this sequence of sums is skewed toward small values of k, such as in the second plot above. In fact, this speeds the random the random tree generator from $O(n^2)$ to $O(n^{3/2})$ time. This in essence, this happens because the Catalan numbers grow so quickly.

For each of the following combinatorial classes $\mathcal{C}$, decide whether it is more efficient to use the *counting order* (1), or the *boustrophedon order* (2) in the design of a fast random object generator for $\mathcal{C} \times \mathcal{C}$. A third possibility is (3)

Each of the orders is essentially as good as the other. (3)

(a) $\mathcal{C}_n$ is the set of permutations of the set $\{1, 2, 3, \ldots, n\}$.

(b) $\mathcal{C}_n$ is the set of ordered triples $(a, b, c) \in \{1, 2, 3, \ldots, n\}^3$.

(c) $\mathcal{C}$ is the class of binary words which do not contain two consecutive zeros (00).

(d) $\mathcal{C}_n$ is the set of labeled unrooted trees with n nodes labeled with $\{1, 2, \ldots, n\}$.