CO 330, FALL 2017, ASSIGNMENT 7

DUE WEDNESDAY NOVEMBER 22 AT 4PM VIA CROWDMARK

PART A

Do all problems in part A.

(1) (a) Give an example of a combinatorial class built of atoms with the property that for each size there are the same number of labelled objects as unlabelled objects of that size.

(b) Give an example of a combinatorial class built of atoms with the property that for each size n, there are $n!$ times as many labelled objects as unlabelled objects of size n.

(c) Give and example of a combinatorial class where the number of labelled objects compared to unlabelled objects is intermediate between the extremes of the previous parts. Plot the first 100 terms of the counting sequence for the labelled and unlabelled version of this class.

(2) The goal of this question is to get an empirical handle on how fast recursive random generation is. This will be done using the example code for binary trees which is on the website. You can use either the maple version or the python version, or you are welcome to re-implement the example in a language of your choice (if you do so please include your code in your solution and make sure you are implementing the same algorithm. It would also be interesting to compare and see what sort of difference it made).

(a) Time the sample code for values of n from 1 to 200 and make a plot. (Show or explain how you obtained your times and your plot) How does the runtime seem to be growing?

(b) Time the sample code a bunch of times for $n = 100$. Plot the times. What does the distribution look like?

To time the code in maple, you can use `st := time(): genTree(n): time() - st;`

PART B

Do **any two** of the following three problems for part B. If you submit more than two only the first two will be graded.

(1) In the unlabelled case the best analogue of the set operator is taking an arbitrary multiset of elements of an unlabelled class $\mathcal{A}$. Write this as $\mathrm{MS{\scriptstyle ET}}(\mathcal{A})$. (A multiset is the best analogue of a set because in the labelled case the labelling forces the pieces to be distinct, but in the unlabelled case you don't have this and so want to allow repeats.)

Suppoes $\mathcal{A}$ is an unlabelled class with $\mathcal{A}_0 = \emptyset$ and let $\mathcal{B} = \textsc{MSet}(\mathcal{A})$, then the ordinary generating functions satisfy the following equation:

$$B(x) = \exp\left(\sum_{k=1}^{\infty} \frac{1}{k} A(x^k)\right)$$

With $A(x) = \sum_{n\geq 1} a_n x^n$, calculate $[x^1]B(x), [x^2]B(x)$, and $[x^3]B(x)$. Explain why these three coefficients are correct for what $\mathcal{B}$ is supposed to be. Why is this formula so much more complicated than the labelled set operator formula?

Proving what the MSET *operator does to generating functions is not so hard, but it is a bit of a digression. For one proof see Flajolet and Sedgewick's "Analytic combinatorics" p29. Another way is with cycle index polynomials.*

(2) Course notes chapter 11 question 9.

(3) Let $\mathcal{T}$ be the class of unlabelled ordered trees with red and blue vertices where each red vertex has at most one red child and any number of blue children, and each blue vertex has an even number of children (with no restriction on colour).

(a) Give a specification for $\mathcal{T}$

(b) Implement a recursive generator (pure recursive, not Boltzmann) for $\mathcal{T}$.

(c) Generate two trees from $\mathcal{T}$ of size $n = 25$ and draw a picture of each of them. Comment on their qualitative shape – are they short or tall, fat or thin, are they similar to each other, etc. (If you write code for the picture drawing please feel welcome to use a larger value of n; 25 was chosen so that people who draw them by hand don't go completely insane.)