Problem for 1997 February

Proposed by Lishan Chen and Dan Jurca

An array $[a_1,\ldots,a_n]$ of, let us say, integers is called a **heap** if:

$$1 \le i \le \lfloor n/2 \rfloor \Rightarrow a_{2i} \le a_i \text{ and } 1 \le i \le \lfloor (n-1)/2 \rfloor \Rightarrow a_{2i+1} \le a_i.$$

(One thinks of a heap as a binary tree with a_1 in the root node; a_2 in the left child node of the root node, a_3 in the right child node of the root node; a_4 and a_5 in the child nodes of the node containing a_2, *etc*. Then a heap is a complete binary tree in which each parent is at least as big as her children.)

Consider a positive integer n, and all n! arrays consisting of data from {1,2,3,…,n} with no repetitions. Some of these arrays are heaps. (For example, if n=6, then the array [6,4,5,2,1,3] is a heap but [5,6,1,2,3,4] is not a heap.) How many of these n! arrays are heaps? Find a method to answer this question. In particular, how many arrays of length 10 (consisting of the data from {1,2,3,…,10} with no repetitions) are heaps? How about the case n=100?

Solution by the proposers

Let h(n) be the number of heaps which can be made from a set of n distinct real numbers. We shall give a recursive formula for h(n). First we recall that a heap is conveniently represented as a complete binary tree-a binary tree with 2^i nodes on level i (the root node is on level 0), except possibly the bottom level; and all the bottom-level nodes are as far to the left as possible. For each positive integer n there is a unique shape for a complete binary tree with n nodes; let l(n) be the number of nodes in the left subtree of the root node of such a tree, and let r(n) be the number of nodes in the right subtree. Then writing $n=2^h+m$ where h is a nonnegative integer and $0 \le m < 2^h$ (h and m are unique), one has

$$l(n)=\begin{cases} 2^{h-1}+m & \text{if } m < 2^{h-1}; \\ 2^h-1 & \text{if } 2^{h-1} \le m. \end{cases}$$

(For example, one finds easily l(10)=6; l(100)=63.) Obviously, r(n)=n–1–l(n).

Now it is immediate that if the array $[a_1,a_2,\ldots,a_n]$ is a heap (consisting of n distinct real numbers), then a_1 is the greatest element of the array, and if the array is arranged in the form of a complete binary tree, then the left subtree of the root node (which contains a_1) and the right subtree of the root node are also heaps. Clearly the elements in the left subtree can be chosen in $((n-1) \| l(n))$ ways; the remaining $r(n)$ elements are in the right subtree. Hence we have the following result:

$$h(n) = \begin{cases} 1 & \text{if } n \le 2; \\ \binom{n-1}{l(n)} \cdot h(l(n)) \cdot h(r(n)) & \text{if } 2 < n. \end{cases}$$

It is now a simple matter to compute, for example, $h(6)=20$, and $h(10)=3{,}360$.

Although we do not have a formula for $h(n)$ in closed form, we can make the following remarks:

i.

If $2 < n$, then either the left subtree (of a complete binary tree with n nodes) or the right subtree is *full*; *i.e.*, has 2^i nodes on level i for $i=0,1,2,\ldots,h$, where h is the *height* (maximum level) of the tree. Since a full binary tree with height h has exactly $2^{h+1}-1$ nodes, and since (for $0 \le n$, $0 \le m \le n$) $(n \| m) = (n \| (n-m))$, it is possible to express the binomial coefficient in the above recursive formula for $h(n)$ so that the ``bottom number'' is 1 less than a power of 2.

ii.

One can always ``reduce'' the evaluation of $h(n)$ to evaluation of a product of binomial coefficients (with ``bottom number'' 1 less than a power of 2) and factors $h(2^k-1)$ for various k. For example, we find (since $l(100)=63$ and $r(100)=36$, *etc*.)

$$\begin{aligned} h(100) &= \binom{99}{63} h(63)h(36) \\ &= \binom{99}{63} h(63) \cdot \binom{35}{15} h(20)h(15) \\ &= \binom{99}{63} \binom{35}{15} h(63)h(15) \cdot \binom{19}{12} h(12)h(7) \\ &= \binom{99}{63} \binom{35}{15} \binom{19}{7} h(63)h(15)h(7) \cdot \binom{11}{7} h(7)h(4) \\ &= \binom{99}{63} \binom{35}{15} \binom{19}{7} \binom{11}{7} h(63)h(15)h(7)^2 \cdot \binom{3}{2} h(2)h(1) \\ &= \left(99 \right) \left(35 \right) \left(19 \right) \left(11 \right) \left(3 \right) h(63)h(15)h(7)^2. \end{aligned}$$

63) (15) (7) (7) (1)

In other words, it is (almost) sufficient to know h(n) where $n=2^k-1$ for some k.

iii.

We do have a formula for h of such n; it follows at once (and is trivially proved by induction) from the recursive formula given above that

$$1 \le k \Rightarrow h(2^k-1)=\prod_{i=1}^{k-1}\binom{2(2^{k-i}-1)}{2^{k-i}-1}^{2^{i-1}}.$$

For example, we find

$$h(63)=\binom{62}{31}\binom{30}{15}^{2}\binom{14}{7}^{4}\binom{6}{3}^{8}\binom{2}{1}^{16};$$

$$h(15)=\binom{14}{7}\binom{6}{3}^{2}\binom{2}{1}^{4};$$

$$h(7)=\binom{6}{3}\binom{2}{1}^{2}.$$

Therefore, continuing with the above example,

$$h(100)=\binom{99}{63}\binom{35}{15}\binom{19}{7}\binom{11}{7}\binom{3}{1}\cdot\binom{62}{31}\binom{30}{15}^{2}\binom{14}{7}^{5}\binom{6}{3}^{12}\binom{2}{1}^{24},$$

evaluation of which product we leave for the reader. □