Let's say we have k colors of items: a_1 of color c_1, a_2 of color c_2, etc. Thus the total number of items in our multiset is $n = a_1 + a_2 + \ldots + a_k$. Likewise, the number of permutations of this multiset is the multinomial coefficient

$$C = \frac{n!}{a_1!a_2!a_3!\cdots a_k!} = \binom{n}{a_1}\binom{n-a_1}{a_2}\binom{n-a_1-a_2}{a_3}\cdots\binom{n-a_1-a_2-\ldots-a_{k-1}}{a_k}.$$

In other words, there are $C_1 = \binom{n}{a_1}$ ways to distribute the a_1 points of color c_1, $C_2 = \binom{n-a_1}{a_2}$ ways to distribute the a_2 points of color c_2 among the remaining $n-a_1$ unoccupied points, and

$$C_i = \binom{n-a_1-a_2-\ldots-a_{(i-1)}}{a_i}$$

ways to distribute the a_i points of color c_i when we get to those, and $C = C_1C_2\cdots C_k$.

Now suppose we select the x_i-th way to distribute the points of color c_i when we get to those. If we index these from zero, this means that $0 \le x_i < C_i$ (where C_i is given by the binomial coefficient above. Then we may index this particular permutation of our original multiset by the expression

$$y = f(x_1, x_2, \ldots, x_k) = x_1 + C_1(x_2 + C_2(x_3 + C_3(\cdots\ldots C_{k-2}x_{k-1}))).$$

Note that, given y in the interval $\{0, 1, 2, \ldots, C-1\}$, we may recover x_1 by dividing y by C_1. The remainder will be x_1 and the quotient will be $x_2 + C_2(X_3 + C_3(\ldots C_{k-2}x_{k-1}))$, in the interval $\{0, 1, 2, \ldots, (C_2C_3\cdots C_k) - 1\}$. Thus we may extract x_2 in a similar manner, and proceed to recover all the x_i's.

In other words, the function f is a bijection from the set of k-tuples $(x_1, x_2, \ldots, x_k)$ (with $0 \le x_i < C_i$ for each i) to the set of integers y in the interval $\{0, 1, 2, \ldots, C-1\}$. We shall use the value y to index our multiset permutations.

What remains is to produce a viable indexing function for the $\binom{m}{j}$ subsets of cardinality j in an m-element set (to produce the x_i's in our function above). We may identify these with binary strings (0's and 1's) of length m in which there are j ones and $m-j$ zeros. We must order those strings, and produce a bijective function which numbers them from 0 to $\binom{m}{j} - 1$.

The standard algorithm would be, successively, to look for the right-most 1 with a zero to the right of it, move it one digit to the right, then pull all other 1's still to the right of it down against it. For example, 00110100111 would change (since the third one can be moved) to 00110011110. If we had three ones and two zeros, we would start with 11100 and proceed as follows. (Note if we complement the strings, we get the binary numbers with two 1's up to length five arranged in the usual order.)

11100	00011
11010	00101
11001	00110
10110	01001
10101	01010
10011	01100
01110	10001
01101	10010
01011	10100
00111	11000

In that case, the function which computes each string's position in the list works like this: To each zero which has 1's to the right of it, associate the binomial coefficient $\binom{n}{m-1}$, where n is the number of digits to the right of it, and m is the number of 1's to the right of it. Add all those numbers. Thus 00110011110 would have the index

$$I = \binom{10}{5} + \binom{9}{5} + \binom{6}{3} + \binom{5}{3} = 408$$

and 00110100111 would have the index

$$I = \binom{10}{5} + \binom{9}{5} + \binom{6}{3} + \binom{4}{2} + \binom{3}{2} = 407.$$

Note, *in the binary case* if we want to index the standard (lexicographic) ordered list on the right, we may simply interchange the role of 1's and 0's in the above calculation, but still starting from the left.

Now we must show how to invert this function, so that our overall indexing function for multiset permutations is invertible. The algorithm for inverting this function is nearly as simple (and quick) as the algorithm for inverting f above. Given the index number I for a string of length m with j ones, note there will be a zero in the first (left) position iff $\binom{n-1}{j-1} \le I$. Thus we may proceed, by a sort of *greedy algorithm* to determine whether each position contains a 1 or a 0. In the latter case, subtract the binomial coefficient from the current index number and proceed.

Algorithm I^{-1}. Given I, m and j:

(1) Let $x = I$, $t = j$ and $i = m$. (Now x, t and i will be altered in the algorithm, but not I, j or m.)

(2) If $\binom{i-1}{t-1} \le x$ then let the $(m+1-i)$-th digit of our string be 0 and let $x = x - \binom{i-1}{t-1}$; otherwise, let the digit be 1 and set $t = t - 1$

(3) Let $i = i - 1$. If $i > 0$, return to step 2; otherwise set all remaining digits to 0 and terminate.

For example, if $I = 407$ and $m = 11$ and $j = 6$, we let $x = 407, i = 11, t = 6$. Now $\binom{i-1}{t-1} = \binom{10}{5} = 252$, which is $\le x$, so we set our first digit to 0, subtract 252 from x (leaving $x = 155$. Now set $i = 10$

Now $\binom{i-1}{t-1} = \binom{9}{5} = 126 \leq 155$, so we set our second digit to 0 and let $x = 155 - 126 = 29$. Let $i = 9$.

Now $\binom{i-1}{t-1} = \binom{8}{5} = 56$ which is *greater than* $x = 29$. So we set our third digit to 1, and let $t = 5$ (we've used up one of the 1's.) Let $i = 8$.

Now $\binom{i-1}{t-1} = \binom{7}{4} = 35$, which still is *greater than* $x = 29$. So we set our fourth digit to 1, and let $t = 4$. Decrement i to 7.

Now $\binom{i-1}{t-1} = \binom{6}{3} = 20 \leq 29$, so we set our fifth digit to 0 and let $x = 29 - 20 = 9$. Decrement i to 6.

Now $\binom{j-1}{i-1} = \binom{5}{3} = 10 > 9$, so we set our sixth digit to 1, and let $t = 3$. Decrement i to 5.

My digits are getting tired, so I'll let the reader finish this example. The output should be 00110100111.

General Case.

Now, by combining the functions f, f^{-1}, I and I^{-1}, we may compute the index of any permutation of our multiset, and from the index, we my recover the permutation.

Muddy example:

Consider the permutation "Mississippi" of the multiset $\{i, i, i, i, m, p, p, s, s, s, s\}$, for example. If we decide to treat the letters in alphabetical order, than $a_1 = 4$ (the i's), $a_2 = 1$ (the M's), $a_3 = 2$ (the p's) and $a_4 = 4$ (the s's). The multinomial coefficient $\binom{11}{4,1,2,4}$ is 34650, so we may ask where in the list, from 0 to 34649, does the permutation "Mississippi" alight? Note that $C_1 = \binom{11}{4} = 330$, $C_2 = \binom{7}{1} = 7$, $C_3 = \binom{6}{2} = 15$ and $C_4 = \binom{4}{4} = 1$.

Now the i's are distributed as 01001001001, which has index

$$\binom{10}{3} + \binom{8}{2} + \binom{7}{2} + \binom{5}{1} + \binom{4}{1} + \binom{2}{0} + \binom{1}{0} = 180$$

(out of the set $\{0, 1, 2, \ldots, 329\}$). Thus $x_1 = 180$.

Next, the M is distributed as 1000000 out of the remaining 7 letters. This gives $x_2 = 0$ (since this is the very first arrangement of seven digits with a single 1 by our algorithm. Note that if we knew x_1 and x_2, we could recover the strings 01001001001 and 1000000, so we could place the letters i and M correctly in the 11-letter string.

Now the p's are distributed as 000011 among the remaining 6 letters, giving $x_3 = 14$, and the s's (of course) will give $x_4 = 0$ since there is only one way to place the last letter, regardless of its multiplicity.

Thus our index for "Mississippi" is $180 + 330(0 + 7(14 + 15(0))) = 32520$ out of the set $\{0, 1, 2, \ldots, 32649\}$.

Having already made the point that if we knew x_1, x_2, x_3 and x_4 we could place all the letters in the word, we need only demonstrate how to recover these x_i's. Begin with the index 32520 and divide by $C_1 = 330$, to obtain a quotient of 98 and remainder of 180. The 180 is our x_1. Now take the quotient 98 and divide it by $C_2 = 7$ to get a quotient of 14 and remainder of 0. The 0 is our x_2. Thus $x_3 = 14$. Note that $x_4 = 0$, always. Done! Invertible indexing. By converting the x_i's into their bit-string equivalents, we can reconstruct Mississippi.

Eine Kleine Nachtmathematik. For the record, if I had to write a full proof of everything, I would include the following combinatorial identity (which is well-known):

$$\sum_{n=k}^{m} \binom{n}{k} = \binom{m+1}{k+1}.$$

It is typically proved by induction, using the Pascal's Triangle identity

$$\binom{m+1}{k+1} = \binom{m}{k} + \binom{m}{k+1}.$$

Thus

$$\begin{aligned}
\binom{m+1}{k+1} &= \binom{m}{k} + \binom{m}{k+1} \\
&= \binom{m}{k} + \binom{m-1}{k} + \binom{m-1}{k+1} \\
&= \binom{m}{k} + \binom{m-1}{k} + \binom{m-2}{k} + \binom{m-2}{k+1} \\
&= \binom{m}{k} + \binom{m-1}{k} + \ldots + \binom{k}{k} + \binom{k}{k+1}.
\end{aligned}$$

But $\binom{k}{k+1} = 0$, completing the proof.