

Re: Combinatorics question

Source: <http://sci.tech-archive.net/Archive/sci.math/2008-04/msg00927.html>

- *From:* Mensanator <mensanator@xxxxxxx>
 - *Date:* Sun, 6 Apr 2008 08:00:32 -0700 (PDT)
-

On Apr 6, 8:21am, Ketakop...@xxxxxxxxxx wrote:

I understood the general method, but I don't quite get the Python algorithm. Let's see, you first create some kind of vector with all combinations of S elements, that is, size 2^S , filled with binaries.

Just a for..loop, no need to track all of them,
just the solutions.

You then enter the method for those with popcount=4.

This is called inside the for..loop, while the binary numbers are created, not afterwards.

But I don't get what gmpy is,

It's an extension module for Python. Third party, not part of the standard distribution. I don't recall the site, but it's easily found by Google.

It's a math library, specifically, the GNU Multi-Precision (GMP) library with a Python wrapper. Highly recommended as it's very fast, supports things Python doesn't have like Rationals & unlimited precision Floats and includes a lot of specialized math functions including the bit manipulations popcount and Hamming Distance. Also does base two conversion which makes the mapping easier.

nor the _combos; is it a string, integer...?

That's the list of numbers that gets created that matches

Re: Combinatorics question

your requested criteria, every one has 4 bits and no pair in the list have a Hamming Distance of 2 or less (meaning no pair has more than 2 matching bits).

The rest of the program maps these binary numbers to the letters 'abcdef', complicated by bit order being reversed.

If you use MSA's corrected algorithm to generate 4-bit numbers, I think you still have to do the Hamming Distance and you may want to note that for mapping purposes I make sure each base 2 string has the appropriate leading 0's so that 15 becomes 001111. That's what the `.zfill(6)` does.

.