Subsets Generation

Suppose we want to generate all the subsets of a given set, then we can use the concept of bit masking to iterate over all the numbers from 0 to 2^N - 1 where N is the total number of elements in the set. Let us try to see this by an example.
Set S = {10,20,30}
We know that the number of subsets of a set having N elements is 2^N. In this case N = 3, so 2^N = 8 subsets.
So taking a binary sequence of length 3, where each bit value represents whether the element of the set is present(bit value is 1) or not(bit value is 0) in the corresponding subset.
0 = 000 => {} - empty subset
1 = 001 => {30}
2 = 010 => {20}
3 = 011 => {20,30}
4 = 100 => {10}
5 = 101 => {10,30}
6 = 110 => {10,20}
7 = 111 => {10,20,30}

Hence, the binary sequence is useful to represent all the subsets of a set.
Pseudocode:

```
function generateSubsets(arr, N)
    /*
        arr represents the set of size N
        Iterating from 0 to 2^N-1 to get all subsets
    */
    for i = 0 to 2^N-1
        /*
            For each i representing a subset checking which
            bits are set in binary representation of i
        */
        for j = 0 to N-1
            if (i & (1 << j))
                print arr[j]
        print newline
    return
```