This exercise is done by Hansong Qi and Zhihao Zhao

Since every subset of a n-set has different characteristic vector, for instance [1, 0, 0] stands for [1], [1, 0, 1] stands for [1, 3], each vector is a binary number of size n, and ranking algorithm is like a conversion from the binary representation to the number.

The code is:

SubsetRank(n, T)

for I from 1 to n do

if i in T then

return r

Similarly, unranking algorithm is like a conversion from number to its binary representation, and use it to generate the subset we want.

The code is:

SubsetUnrank(n, r)

for i from n to 1 do

if r mod 2 = 1 then

r <- r/2(ground down)

return T

For a given subset, the successor algorithm needs to give us its next subset, to do that we can think of increment of a binary number.

The code is:

Successor(n, T):#Successor algorithm

while () and () do:

if hen

return T

Since we have the successor algorithm, we can use it to generate a collection of subsets of n-set. We know that a set of cardinality n has 2^n subsets, therefore we need to run the successor algorithm for 2^n times in order to find all subsets.

The code is:

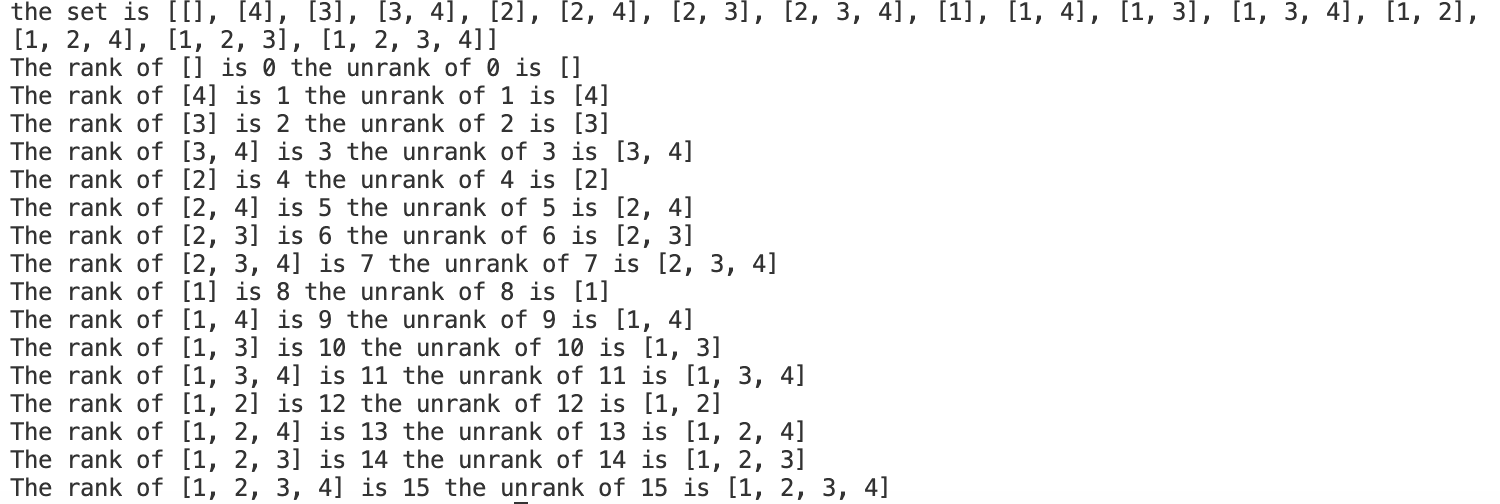
Collection(n):#Generate all subsets of n\_set using successor algorithm

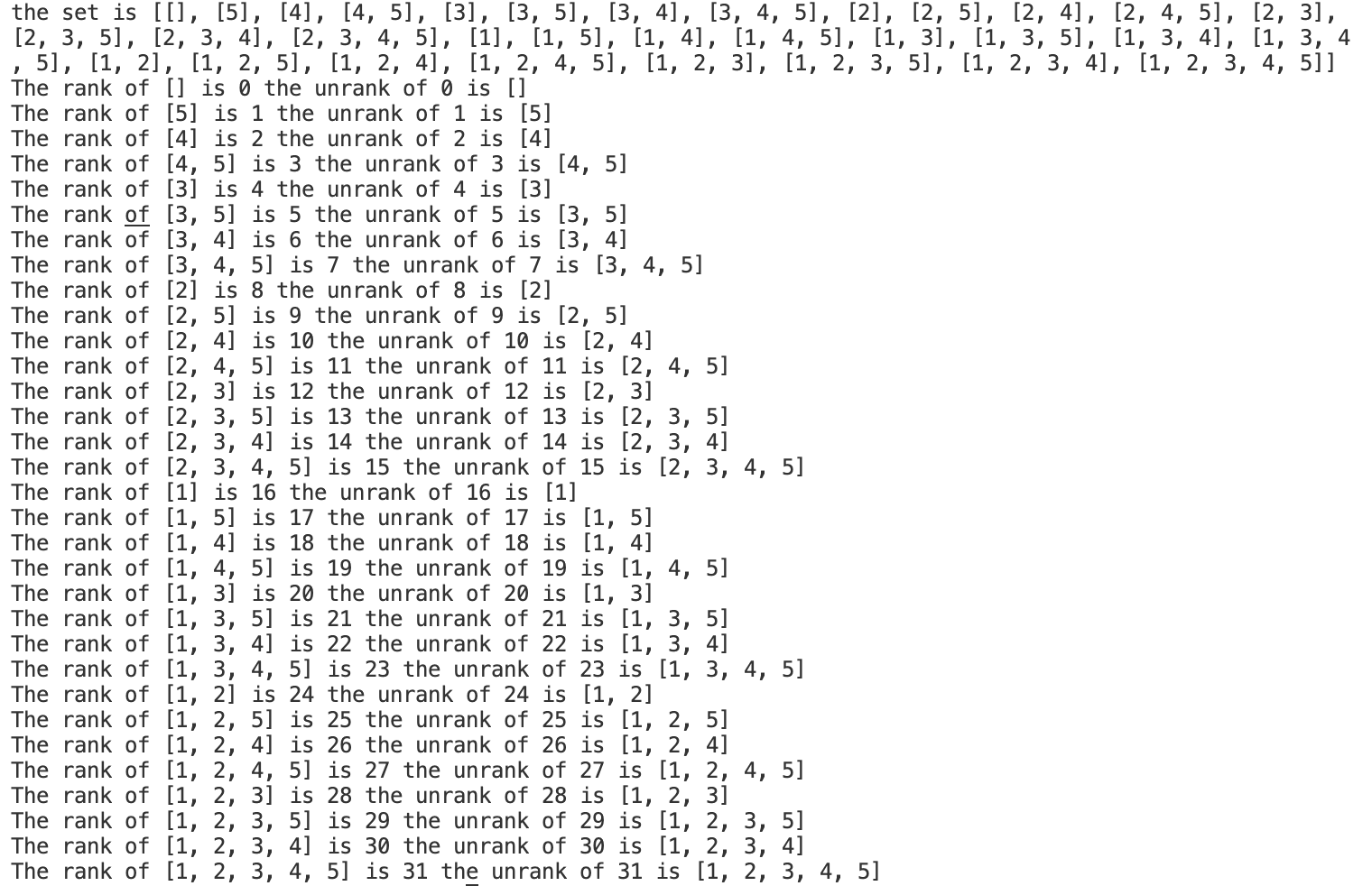
for I from 1 to :

subset Successor(n, subset)

subsets subsets + {subset}

return subsets

Result for n = 4 is: 

Result for n = 5 is: 

Result for n = 6 is: 