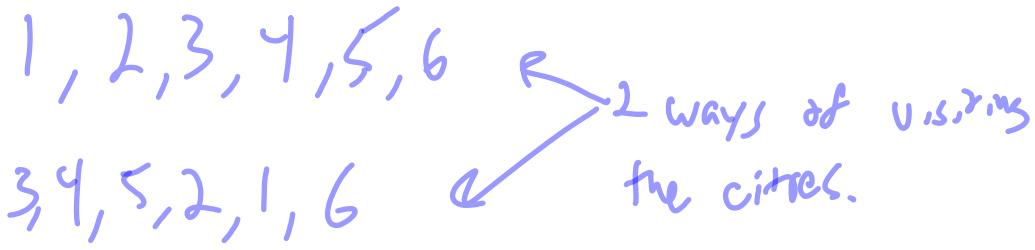
In some situations, it is not enough to be able to count the total number of permutations/combinations, and instead we need to be able to generate each possible permutation or combination.

Suppose a salesman needs to visit 6 different cities as efficiently as possible. Each permutation of the cities gives a different order in which the cities can be visited.



Suppose a company has 100 employees and needs to choose a committee of 15 employees such that the employees together cover 25 needed skills (an employee may have more than one skill). Each 15-combination of the 100 employees is a different candidate solution which we can then check to see if the 25 skills are covered.

Generating Permutations: Suppose we want to generate all of the permutations of a set of n elements.

There will be $n!$ total permutations.

Lexicographic ordering of the permutations:

$a_1 a_2 a_3 \cdots a_n$ and $b_1 b_2 b_3 \cdots b_n$

The first permutation comes before the second if for some K,

$a_1 = b_1$, $a_2 = b_2$, $\cdots$, $a_{K-1} = b_{K-1}$, and $a_K < b_K$.

Example: 1 2 3 and 1 3 2. $K=2$ because $a_1 = b_1$ and $a_2 \neq b_2$.

Since $a_2 < b_2$, then 123 comes before 132 in lexicographical order.

Smallest permutation? $1, 2, 3, \ldots, n$

Largest permutation? $n, n-1, n-2, \ldots, 1$

We will save an algorithm for generating permutations starting from the smallest and ending at the biggest.

$1, 2, 3, \ldots, n-1, n \rightarrow 1, 2, 3, \ldots, n-2, n, n-1 \rightarrow 1, 2, \ldots, n-3, n-1, n-2, n$

$\rightarrow 1, 2, 3, \ldots, n-3, n-1, n, n-2 \rightarrow 1, 2, 3, \ldots, n-3, n, n-2, n-1$

Find index a_j and a_{j+1} with $a_j < a_{j+1}$ and $a_{j+1} > a_{j+2} > a_{j+3} > \cdots > a_n$.

In position j: place the smallest number from $a_{j+1}, \ldots a_n$ that is greater than current a_j. List remaining numbers in increasing order.

Example: What is the next permutation in lexicographic order after 362541?

$a_j = 2$ as $2 < 5$ and $5 > 4 > 1$

In the next permutation in position j, we place the smallest number from 5,4,1 that is bigger than a_j. new $a_j = 4$

Remains elements are placed in increasing order.

Next permutation: 364125

Example: Generate the permutations of the integers 1,2,3 in lexicographic order.

123

132

213

231

312

321

Generating Combinations: How can we generate all combinations of a set of n elements? That is, we are interested in all 0-combinations, 1-combinations, ..., and n-combinations.

We can use bit strings to generate combinations.

Suppose we have 4 elements: _ _ _ _

Each combination can be represented as a bit string where we place a 1 in position i if object i is in the combination or a 0 otherwise.

Ex: 0 1 1 0 $\Rightarrow$ Combination with objects 2 & 3

We can generate all combinations by generating all bit strings

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

Now suppose we only want to generate r-combinations (i.e. we only want to generate combinations of size r). Can we do better?

We still use a bit string representation but we only want to consider strings that have exactly r 1's in them.

Suppose we want to consider 2-combinations of 4 elements.

1 1 0 0

1 0 1 0

1 0 0 1

0 1 1 0

0 1 0 1

0 0 1 1

We want to generate all ways of arranging the 0's and the 1's. Note the 1's are indistinguishable and 0's

List all 3-combinations of 1,2,3,4,5.

1,2,3

1,2,4

1,2,5

1,3,4

1,3,5

1,4,5

2,3,4

2,3,5

2,4,5

3,4,5

1 1 1 0 0

1 1 0 1 0

1 1 0 0 1

1 0 1 1 0

1 0 1 0 1

1 0 0 1 1