



Ask a Question



Introduction to Subsets

Let's go over the Subsets pattern, its real-world applications, and some problems we can solve with it.

We'll cover the following... ▼

About the pattern

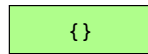
The **subsets** pattern is an important strategy to solve coding problems that involve exploring all possible combinations of elements from a given data structure. This pattern can be useful when dealing with sets containing unique elements or arrays/lists that may contain duplicate elements. It is used to generate all specific subsets based on the conditions that the problem provides us.

The common method used is to build the subsets incrementally, including or excluding each element of the original data structure, depending on the constraints of the problem. This process is continued for the remaining elements until all desired subsets have been generated.

The following illustration shows how subsets are made from a given array:

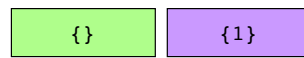
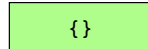


We start with an empty set, iterate through all elements, and add them to all existing sets to create subsets.



1 of 4

Set up the first two subsets: the empty set and the set with just the first element, 1.

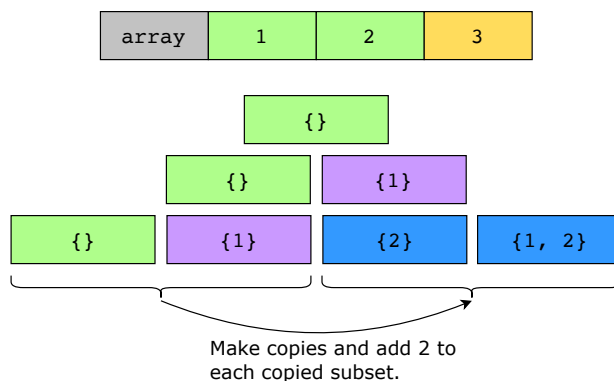


Make copies and add 1 to each copied subset.

2 of 4

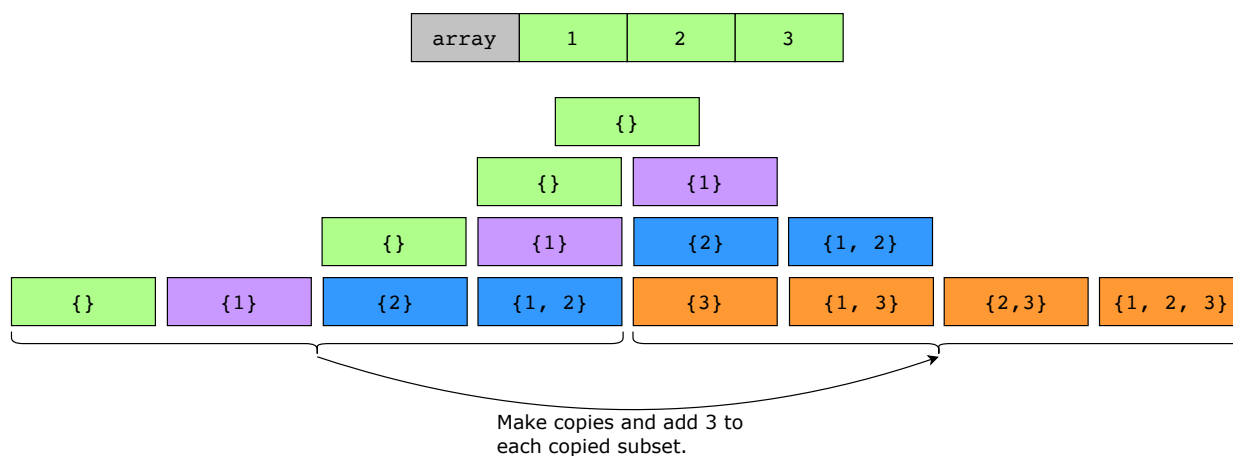


Now, let's add the second element, 2, to each of the subsets we've already created. This yields {2} and {1, 2}.



3 of 4

Now, let's add the third element, 3, to each of the subsets we've already created. This yields {3}, {1, 3}, {2, 3} and {1, 2, 3}.



4 of 4

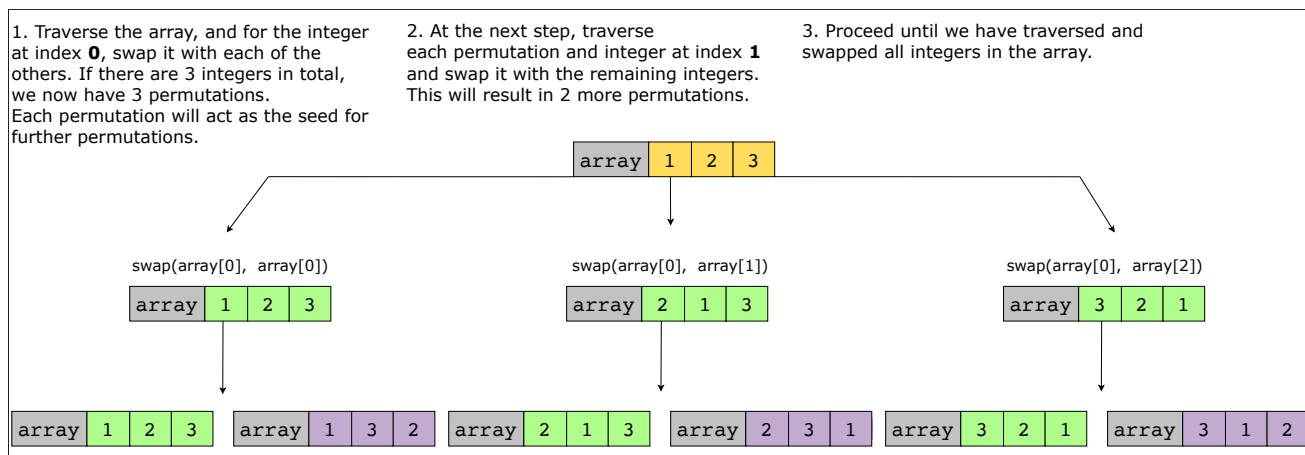
Note: We sometimes also use a programming technique known as backtracking to generate the required subsets of a given data structure of elements. Backtracking applies to a broader range of problems where exhaustive search, that is, evaluating all possibilities, is required. These problems may involve various

constraints, rules, or conditions that guide the exploration process. Not all of these problems involve finding subsets. That is why it is necessary to discuss Subsets as a separate programming pattern.

Examples

The following examples illustrate some problems that can be solved with this approach:

1. **Permutations:** Return all possible permutations of an array of distinct integers.



2. **Combination sum:** Return all combinations of integers in an array that add up to a target number.

?

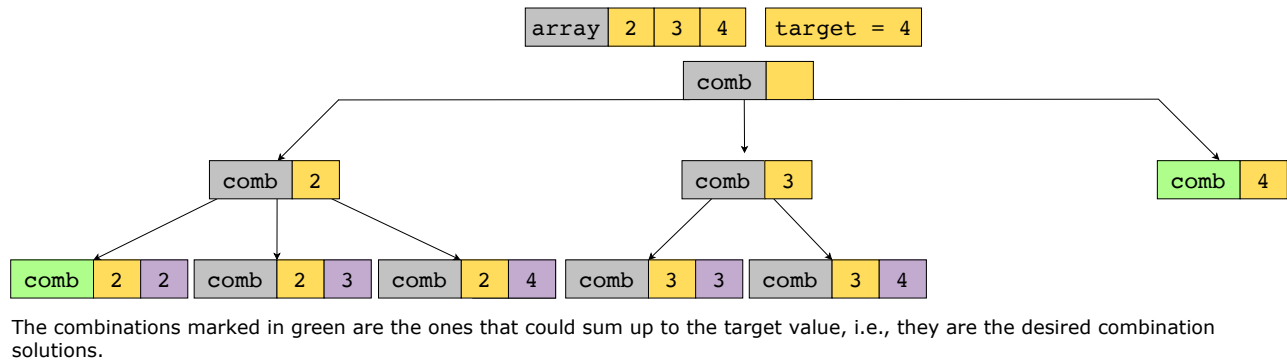
Tt



Step 1: Traverse the array, pick each integer from the input array, and add it to the combinations.

Step 2: Traverse each combination, and if the sum of the combination equals the target, save this combination. If it is less than the target, move to step 3. If the new sum exceeds the target, explore other combinations.

Step 3: Pick each integer from the input array and add it to the combination and repeat step 2.



Does your problem match this pattern?

Yes, if the following condition is fulfilled:

- **Requirement for combinations or subsets:** The problem asks us to generate combinations (or subsets) of the elements from the input data structure. This could be the final solution itself or a step toward reaching the solution.

Real-world problems

Many problems in the real world use the subsets pattern. Let's look at some examples.

- **Custom movie playlists:** Using the subsets pattern, we can generate all possible combinations of movies to meet any viewer's preference. Imagine we have a list of movies and want to create custom playlists based on different criteria, such as genre or length ? For example, generating all combinations of comedies and dramas for a weekend binge-watching session.

T



- **Test case generation:** In software testing, particularly in automated testing, generating comprehensive test cases is crucial. Using the subsets pattern, we can generate various combinations of input values to test different scenarios and edge cases thoroughly.
- **Feature selection in machine learning:** In machine learning, feature selection plays a vital role in building effective models. By considering subsets of features from a dataset, we can evaluate the performance of different combinations of features and select the subset that optimizes model performance and complexity.

Strategy time!

Match the problems that can be solved using the subsets pattern.

Note: Select a problem in the left-hand column by clicking it, and then click one of the two options in the right-hand column.

Match The Answer

ⓘ Select an option from the left-hand side

Find the longest possible route of 1s in a matrix of 0s and 1s.

Subsets

Find any one subset in a

Some other pattern

?

Tt

