Python - Backtracking

Backtracking is a form of recursion. But it involves choosing only option out of any possibilities. We begin by choosing an option and backtrack from it, if we reach a state where we conclude that this specific option does not give the required solution. We repeat these steps by going across each available option until we get the desired solution.

Below is an example of finding all possible order of arrangements of a given set of letters. When we choose a pair we apply backtracking to verify if that exact pair has already been created or not. If not already created, the pair is added to the answer list else it is ignored.

Example

```
def permute(list, s):
   if list == 1:
      return s
   else:
      return [
         y + x
         for y in permute(1, s)
         for x in permute(list - 1, s)
      ]
print(permute(1, ["a","b","c"]))
print(permute(2, ["a","b","c"]))
```

Output

When the above code is executed, it produces the following result −

```
['a', 'b', 'c']
['aa', 'ab', 'ac', 'ba', 'bb', 'bc', 'ca', 'cb', 'cc']
```

Print Page