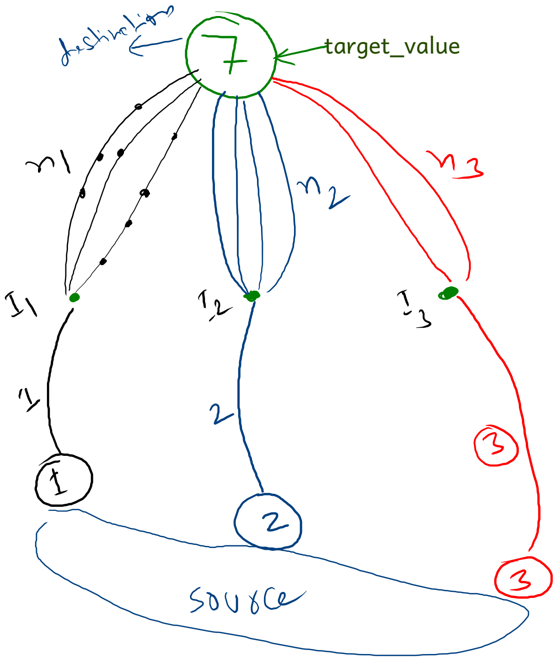
Find Permutation of given numbers whose sum is equal to a target_value

Example problem statement :

Find all the possible stair path with given total number of stairs step and allowed step size. e.g.

Total number of setps = 7 and at a time we can take step of size 1 or 2 or 3.



Total number of paths :n1 + n2 + n3

n1: initial step = 1unit; distance of   'I1 to 7' = 6 unit

n2: inital step = 2unit; distance of 'I2 to 7' = 5 unit

n3 : inital step = 3unit; distance of I3 to 7 = 4 unit

```
private List<String> getStairPathPermutation1(int targetValue, int... allowedSteps) {
    // targetValue becomes negative for invalid path
    if (targetValue < 0) {
        return List.of();

    }
    // targetValue becomes 0 for valid end of path
    if (targetValue == 0) {
        return List.of("");
    }

    List<String> paths = new ArrayList<>();
    for (int i = 0; i < allowedSteps.length; i++) {
        List<String> ipaths = getStairPathPermutation1(targetValue - allowedSteps[i], allowedSteps);

        for (String path : ipaths) {
            paths.add(allowedSteps[i] + path);
        }
    }
    return paths;
}
```