Algorithm for generating all derangements on \( \{1,2,...,n\} \)

**Algorithm:** GenerateDerangement(n)

**procedure** RecGenDer(n)

1. D[n]=[]; //initialize the list of derangement on n points to empty list
2. for a from 1 to n-1
   1. b[n]=a // b is a derangement
   2. for each permutation pi in D[n-1] // run through all permutations in the list D[n-1]
      1. // define the new permutation based on each permu pi
      2. b[c] = pi[c] whenever pi[c] is not equal a
      3. b[c] = n if pi[c]=a
      4. Append the permutation in to the list D[n]
   3. For a from 1 to n-1
      1. B[n] = a
      2. B[a] = n
      3. For each permutation pi in D[n-2]
         1. B[c]=pi[c] if c<=n-1 and c<>a and pi[c]<>a
         2. B[n-1]= pi[a]
         3. B[c]=n-1 if pi[c]=a
         4. Append the permutation in to the list D[n]

**Main**

1. D[2]=[[2,1]]
2. D[3]=[[2,3,1],[3,1,2]]
3. If n<4, then return D[n]
4. Else for j from 4 to n
   1. RecGenDer(j)
5. Return(D[n])