A few more examples w/ recursion.

Permutations (bijective functions (1-1, onto) from a set to itself.)
(finite)

Another way to think of it is all re-orderings of a list.

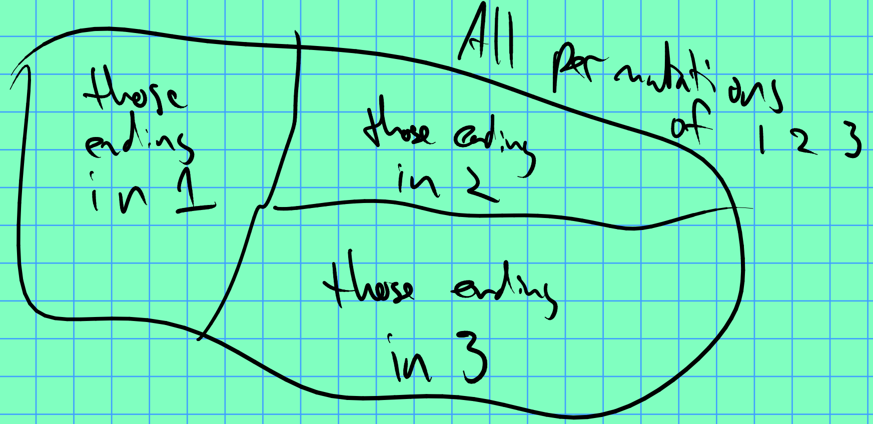
Ex: Say list = 1 2 3.

Then permutations of 1 2 3 are

1 2 3
2 1 3
1 3 2
3 1 2
2 3 1
3 2 1

for a list of size n, there are n! permutations.

Idea: for every element, give it a turn to be last, and permute the first n-1 elements in all possible ways.



Base case? Say n=1 (n = size of list)

L = [1]

Then Perms(L) = [[1]].

Sketch in C++: (n = L.size())

```
vector <vector <int>> perms (vector <int> L)
{
      if (L.size() == 1) {
          vector<vector <int>> P;
          P.push_back(L);
          return P;
      }
    int n = L.size();
   vector <vector <int>> P; // ret. val.
   for (int i = 0; i < L.size() i++) {
        // put L[i] last.
       swap(L[i], L[n-1]);
       int x = L[n-1];
      L.pop_back();
     vector<vector<int>> T = perms(L);
     for (j = 0; j < T.size(); j++) {
            T[j].push_back(x);
            P.push_back(T[j]);
     }

   L.push_back(x);
   swap(L[i], L[n-1]);
```

```
    }
    return p;
}
```