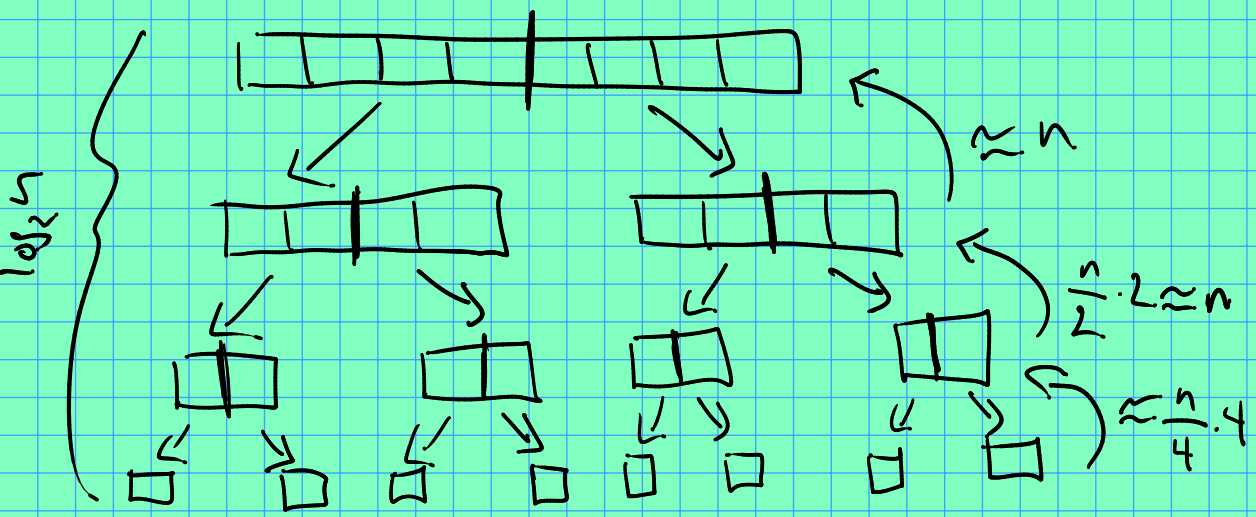
Merge Sort:

Exercise: review "selection sort".

Idea: ① Divide Array in 2 equal pieces.
② Recursively sort each one.
③ Recombine results by "merging" the sub arrays back together.



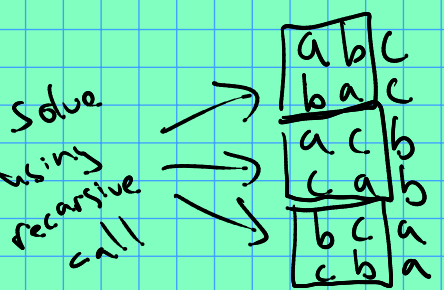
Total time: $\approx n \cdot \log_2 n$.

$$n \to \frac{n}{2} \to \frac{n}{4} \cdots \frac{n}{2^l} = 1$$

$$n = 2^{\textcircled{l}}$$

$$\log_2 n = l.$$

Permutations: on an input string, e.g. "abc", output all permutations:



```
// base case:
if (s == "") return vector<string>(1, "");
```

// Idea:

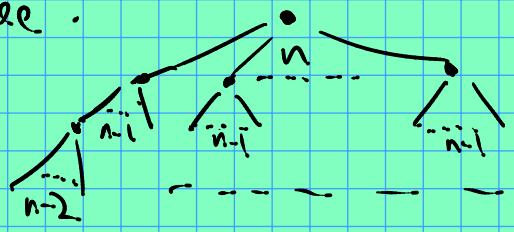
for i = 0, ... s.length - 1,

① swap S[i] ⟷ S[length - 1]

② call permS(S[0, ... length - 2])

③ for each entry in ↖, add last element back.

Recursion tree:



Tree for MergeSort: