Combinatorial Objects

➪ **Index set**

? ➪ **Power set**

Items from a finite set may be combined in many ways to create interesting objects which can be expressed as sequences of item indices.

➪ Sets of objects

Index set {1,2,3}

➪ 2 generation questions

✎ List set members: set permutations

123 132 312 321 231 213

✎ Group set members: subsets

ϕ {1} {2} {3} {1,2} {1,3} {2,3} {1,2,3}

Combinatorial Objects Generating

- Minimal-change order
- Lexicographic order

Many problems can only be solved in general by going through some combinatorial objects.

Fundamental problem

Issues to consider

Do we need to do more than going through the required objects ?

- How to generate?

Do we need the objects in a specific order?

- In what order?

Examples: set permutations

Minimal-change order: 123 132 312 321 231 213

Lexico-graphic order: 123 132 213 231 312 321

Combinatorial Objects
Algorithms Overview

Long story short: iterate on (sub)instance size

Simple but require more work than necessary to generate the objects, suitable perhaps for pen-paper application.

➪ Straight decrease-conquer

- ☑ Permutations: minimal-change
- ☑ Subsets (what order?)

Do not require the extra work of generating smaller objects.

➪ Better procedures

- ☑ Minimal-change permutations
- ☑ Lexicographic permutations
- ☐ Subsets from bit vectors

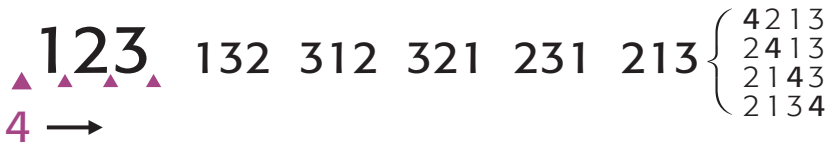
Generating Algorithms
Decrease-by-One

➪ **Squashed order**

It's easy to generate permutations of size n given those of size n–1.

Exercise
Insert 4 in permutations of {1,2,3}.

➪ Key observation



➪ Bottom-up decrease-by-1

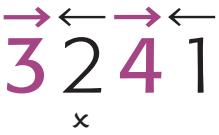
Quiz
Why is it useful to alternate (change) insert direction in bottom-up permutation generation?

➪ Evaluation

Generation order, efficiency hit

Generating Permutations
Johnson-Trotter

➪ Mobile element

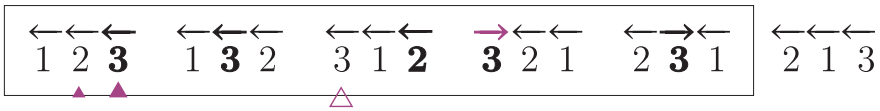


Quiz
What's a mobile element in a permutation?

Algorithm *JohnsonTrotter*(n)
Input Integer $n > 0$
Output All permutations of $\{1, 2, \ldots, n\}$

1: initialize, output first permutation $\{\overleftarrow{1}, \overleftarrow{2}, \ldots, \overleftarrow{n}\}$
2: **while** last permutation has mobile elements **do**
3: find largest mobile element k such that $j\ \overleftarrow{k}$
4: swap k, j
▷ 5: $\forall$ elements $> k$, reverse direction
6: add new permutation to output list

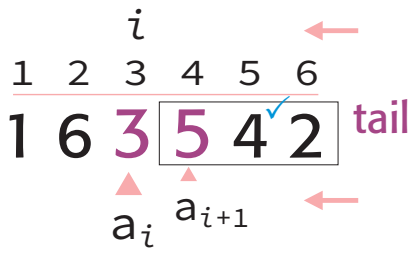
Quiz
In what order does J-T generate the permutations?



Generating Permutations
Lexicographic Order

For next lexicographic permutation

- Scan from right until first pair $a_i < a_{i+1}$
- Swap a_i with smallest tail element > a_i
- Arrange tail elements in increasing order



1 2 3
123
132
213
231
312
321

Check textbook for special procedure to solve n=3 instance of problem.

1234
1243
...
1432
? 2134 ◀
2143
...

Exercise
Use procedure to complete the list, verify in *WolframAlpha*, type:
`permutations of {1,2,3,4}`

Generating Subsets
A Better Method

- **Bit vectors**
- **Reflected Gray code**

Subsets from bit vectors

Quiz
Which subset of {a,b,c} is encoded by 001?

- Encoding subsets
- How to generate?

Check homework exercises.

- Squashed order (hint)

Quiz
What's the name of the binary code that gives minimal-change listing of subsets?

- Minimal-change order

Exercise
Write a pseudocode for an exhaustive search solution of the Knapsack problem where subsets are coded as bit vectors. Which order would be preferred, if any?