

Surabhi Narayan

Department of Computer Science & Engineering



# **DECREASE AND CONQUER**

## Surabhi Narayan

Department of Computer Science & Engineering

#### **Decrease and Conquer**

# **Combinatorial Objects**

- > Permutations
- **Combinations**
- > Subsets of a given set



#### **Decrease and Conquer**

# **Generating Permutations**

- ➤ Underlying set elements are to be permuted
- ➤ Decrease and conquer approach
- > Satisfies the minimal change requirement
- Example: Johnson- Trotter algorithm



#### **Decrease and Conquer**

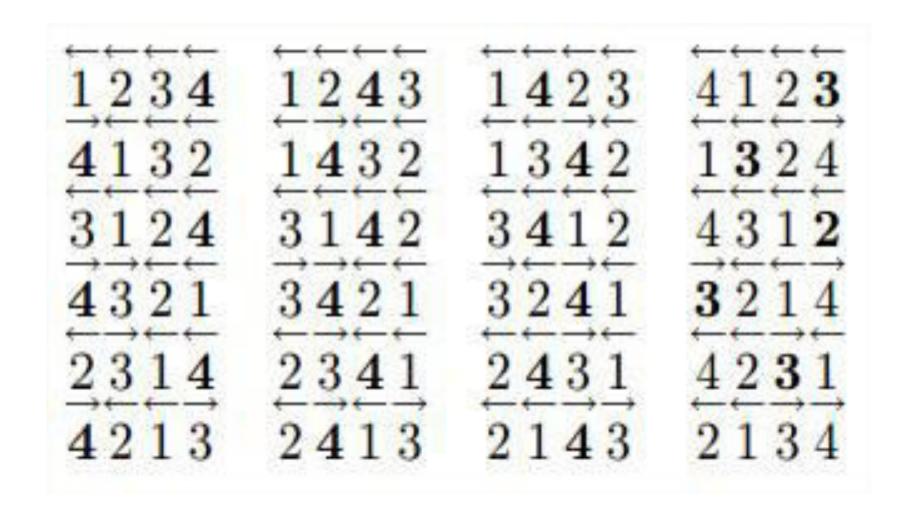
# **Generating Permutations**

#### **ALGORITHM** JohnsonTrotter(n)

```
//Implements Johnson-Trotter algorithm for generating permutations
//Input: A positive integer n
//Output: A list of all permutations of \{1, \ldots, n\}
initialize the first permutation with 1 \ 2 \ldots n
while the last permutation has a mobile element do
find its largest mobile element k
swap k with the adjacent element k's arrow points to
reverse the direction of all the elements that are larger than k
add the new permutation to the list
```



## **Decrease and Conquer**





#### **Decrease and Conquer**



## **Generating Subsets:**

Knapsack problem needed to find the most valuable subset of items that fits a knapsack of a given capacity.

Powerset: set of all subsets of a set. Set  $A=\{1, 2, ..., n\}$  has  $2^n$  subsets.

Generate all subsets of the set  $A=\{1, 2, ..., n\}$ .

```
Any decrease-by-one idea?

# of subsets of \{ \} = 2^0 = 1, which is \{ \} itself

Suppose, we know how to generate all subsets of \{ 1,2,...,n-1 \}

Now, how can we generate all subsets of \{ 1,2,...,n \}?
```

#### **Decrease and Conquer**



## **Generating Subsets:**

```
All subsets of \{1,2,...,n-1\}: 2^{n-1} such subsets
```

```
All subsets of \{1,2,...,n\}:
 2^{n-1} subsets of \{1,2,...,n-1\} and
 another 2^{n-1} subsets of \{1,2,...,n-1\} having 'n' with them.
```

That adds up to all  $2^n$  subsets of  $\{1,2,...,n\}$ 

#### **Decrease and Conquer**



## **Alternate way of Generating Subsets:**

Knowing the binary nature of either having **n**th element or not, any idea involving binary numbers itself?

One-to-one correspondence between all  $2^n$  bit strings  $b_1b_2...b_n$  and  $2^n$  subsets of  $\{a_1, a_2, ..., a_n\}$ .

Each bit string  $b_1b_2...b_n$  could correspond to a subset.

In a bit string  $b_1b_2...b_n$ , depending on whether  $b_i$  is 1 or 0,  $a_i$  is in the subset or not in the subset.

000	001	010	011	100	101	110	111
Ø	$\{a_3\}$	$\{a_2\}$	$\{a_2, a_3\}$	$\{a_1\}$	$\{a_1, a_3\}$	$\{a_1, a_2\}$	$\{a_1, a_2, a_3\}$

#### **Decrease and Conquer**



## Generating Subsets in Squashed order:

**Squashed order:** any subset involving  $a_j$  can be listed only after all the subsets involving  $a_1, a_2, ..., a_{j-1}$ 

Both of the previous methods does generate subsets in squashed order.

000 001 010 011 100 101 110 111 
$$\varnothing$$
 { $a_3$ } { $a_2$ } { $a_2$ ,  $a_3$ } { $a_1$ } { $a_1$ ,  $a_3$ } { $a_1$ ,  $a_2$ } { $a_1$ ,  $a_2$ ,  $a_3$ }

#### **Decrease and Conquer**



## Generating Subsets in Squashed order:

**Squashed order:** any subset involving  $a_j$  can be listed only after all the subsets involving  $a_1, a_2, ..., a_{i-1}$ 

Can we do it with minimal change in bit-string (actually, just one-bit change to get the next bit string)? This would mean, to get a new subset, just change one item (remove one item or add one item).

## Binary reflected gray code:

000 001 011 010 110 111 101 100



## **THANK YOU**

Surabhi Narayan
Department of Computer Science & Engineering
surabhinarayan@pes.edu