

Today's Content

1. Printing Next Permutation
2. Wave array.

Permutations: #Diff arrangement

1: c e a

2: c a e

3: e a c

Write permutations

4: e c a in Lexicographical

5: a e c

6: a c e

1: a c e

2: a e c

3: c a e

4: c e a

5: e a c

6: e c a

Q Given an arrangement, next permutation in Lexicographical order

Dictionary Order

Given 10 14 5 : Next Permutation Lex: 14 5 10

Given 14 5 10 : Next Permutation Lex: 14 10 5

Given 14 10 5 : Next Permutation Lex: -1;

Ex:

All permutations of 10 14 5

5 10 14

5 14 10

10 5 14

10 14 5

14 5 10

14 10 5

Given an array find the next permutation of array in Dictionary Order

Ex1: $\text{arr}[] = \{1 \ 2 \ 3 \ 4\}$ #Output: $\{1 \ 2 \ 4 \ 3\}$

Ex2: $\text{arr}[] = \{2 \ 1 \ 3\}$ #Output: $\{2 \ 3 \ 1\}$

Ex3: $\text{arr}[] = \{4 \ 3 \ 2\}$ #Output: $\{-1\}$

#Obs: If Data is decreasing order, no next permutation

Ex4: $\text{arr}[] = \{8 \ 10 \ 7 \ 6 \ 5\} \rightarrow \{10 \ 8 \ 7 \ 6 \ 5\} \xrightarrow{\uparrow \text{ reverse it}} \{10 \ 5 \ 6 \ 7 \ 8\}$

#Obs1: After 8, data decreasing, means with 8, we cannot start any other permutations

#Obs2: Swap 8 with smallest element greater than 8

Ex5: $\text{arr}[] = \{7 \ 12 \ 9 \ 4 \ 2 \ -1\} \rightarrow \{9 \ 12 \ 7 \ 4 \ 2 \ -1\} \xrightarrow{\uparrow \text{ reverse it}} \{9 \ -1 \ 2 \ 4 \ 7 \ 9\}$

Ex6: $\text{arr}[] = \{9 \ 18 \ 15 \ 12 \ 10 \ 6 \ 4 \ 2 \ 0\}$

$\xrightarrow{\uparrow \text{ reverse it}}$
 $\{10 \ 18 \ 15 \ 12 \ 9 \ 6 \ 4 \ 2 \ 0\}$

$\xrightarrow{\uparrow \text{ reverse it}}$
 $\{10 \ 0 \ 2 \ 4 \ 6 \ 9 \ 12 \ 15 \ 18\}$

Ex7 $\text{arr}[] = \{3 \ 5 \ 11 \ 7 \ 20 \ 18 \ 14 \ 10 \ 6 \ 3 \ 1\}$

$\begin{matrix} 3 & 5 & 11 & 7 & 20 & 18 & 14 & 10 & 6 & 3 & 1 \\ \times & \times & \times & \checkmark & & & & & & & \end{matrix}$

reverse it

$\begin{matrix} 3 & 5 & 11 & 10 & 20 & 18 & 14 & 7 & 6 & 3 & 1 \\ \times & \times & \checkmark & & & & & & & & \end{matrix}$

reverse it

Ex8

$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

$\text{arr} = \{ 19, 14, 7, 6, 20, 18, 13, 12, 10, 4, 2 \}$

#Step1: Find index i , such that $[i+1..N-1]$ data is decreasing?

if $i = N-2$:

while($\text{arr}[i] > \text{arr}[i+1]$):

$i--$

if($i == -1$): #No next permutation return -1;

return $\text{ans}[i:N-1]$;

3 return ans ;

$\text{arr} = \{ 8, 6, 4, 3 \}$

$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

#Step2: Find ceil of $\text{arr}[i]$, $i \in [i+1..N-1]$

int $j = N-1$;

while($\text{arr}[i] > \text{arr}[j]$):

$j--$

$\text{arr} = \{ 19, 14, 7, 6, 20, 18, 13, 12, 8, 9, 10 \}$

#Step3: Swap $\text{arr}[i]$ & $\text{arr}[j]$;

$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

#Step4: Reverse $\text{arr}[]$ from $i+1..N-1$;

$\text{arr} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

$\text{arr} = \{ 19, 14, 7, 10, 2, 4, 6, 12, 13, 18, 20 \}$

vector<int> Permutation(vector<int> &arr) {
 TC: $O(3N+1) = O(N)$

28 Wave Array

Given $ar[n]$ elements re-arrange array in a wave form such that $a[0] \geq a[1] \leq a[2] \geq a[3] \leq a[4] \geq a[5] \geq \dots$

Note: If multiple right answers are possible return any one of them.

0 1 2 3 4 5

Ex1: $ar[] = \{5, 10, 12, 14, 20, 24\}$

Output1 = $\{10, 5, 14, 12, 24, 20\}$ # Note: All of them are valid.

Output2 = $\{12, 5, 14, 10, 24, 20\}$

Output3 = $\{12, 5, 24, 10, 20, 14\}$

0 1 2 3 4 5 6

Ex2: $ar[] = \{7, 9, 10, 14, 9, 8, 7\}$

Output = $\{9, 7, 10, 8, 14, 7, 9\}$

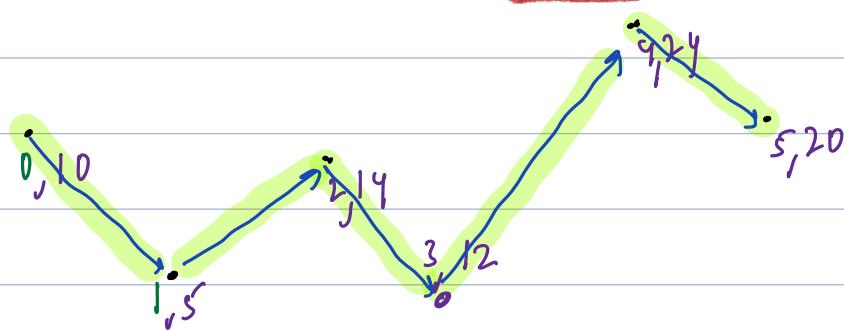
Ideal 1. Sort $ar[]$

2. Swap pair of elements $ar[i]$ & $ar[i+1]$ & update $i = i + 2$

0 1 2 3 4 5

$ar[] = \{5, 12, 20, 14, 10, 24\}$

$ar[] = \{5, 10, 12, 14, 20, 24\}$



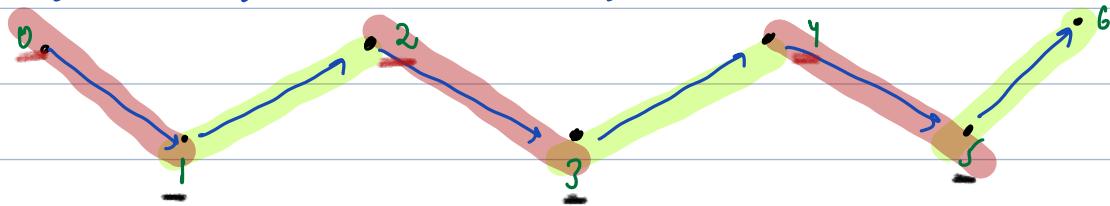
TC: $O(N \log N)$ SC: $O(1)$

Obs: Sorting approach will give us lexicographically 1st wave array.

Idea 2:

Expected arrangement:

$ar[0] \geq ar[1] \leq ar[2] \geq ar[3] \leq ar[4] \dots$



Observing

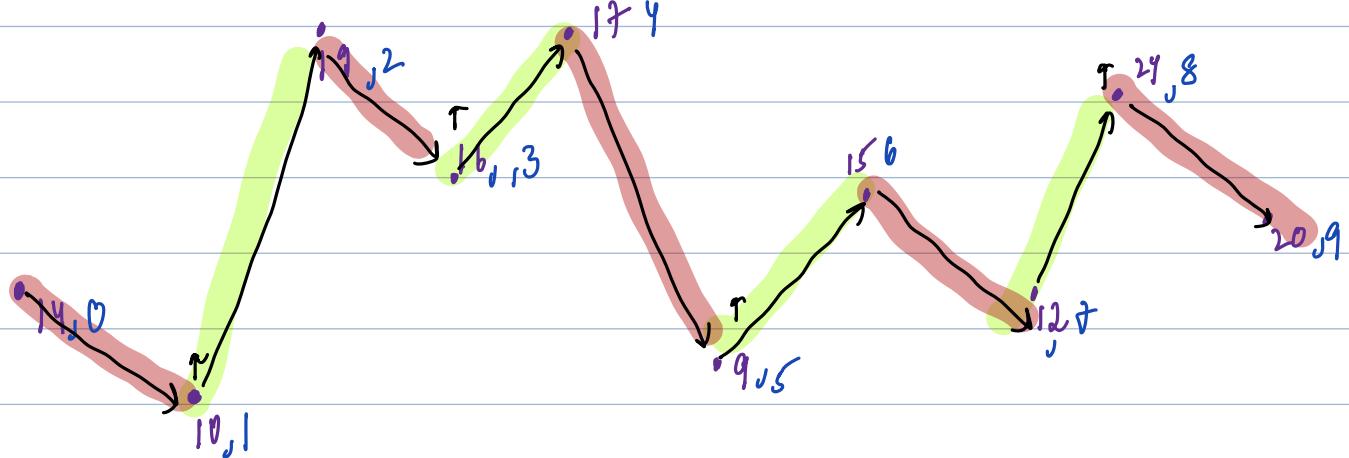
$\{ i \mid a[i] = 0 \}$

Ideally $ar[i] > ar[i+1]$

$\{ i \mid a[i] = 1 \}$

Ideally $ar[i] \leq ar[i+1]$

Ex2: $ar[] = \{ 10, 14, 19, 17, 16, 19, 15, 12, 18, 20, 24, 20 \}$



int

vector<int> waveArray(vector<int> arr) { TC: O(N) SC: O(1) }

```
int N = arr.size();
for(int i=0; i<N; i++) {
    if(i%2 == 0) {
        if(arr[i] < arr[i+1]) { swap(arr[i], arr[i+1]); }
    } else {
        if(arr[i] > arr[i+1]) { swap(arr[i], arr[i+1]); }
    }
}
return arr;
```

3