

## Prefix/Range Sum/Max/Min (no update)

---

Range Sum Query 1D - Immutable `B = AB-A`

Range Sum Query 2D - Immutable `D = ABCD+A-AB-AC` `rowwise & colwise`

Product of Array Except Self `preProduct, postProduct`

Partition Array into Disjoint Intervals `preMax, postMin`

Trapping Rain Water I `preMax, postMax` `bfs: heap[boundary], max_visited`

Trapping Rain Water II `bfs: heap[boundary], max_visited`

## Binary Indexed Tree (prefix sum with update)

---

Range Sum Query 1D - Mutable `BIT`

Range Sum Query 2D - Mutable `BIT-2D`

Count of Smaller Numbers After Self `BIT` `num2ind` `ind2ind[i]=i-1`

Reverse Pairs (Count of Twice Smaller Numbers After Self)

`BIT` `ind2num=sorted(nums)` `num2ind` `ind2ind: binary search`

```

"""Binary Indexed Tree"""
class BinaryIndexedTree(object):
    def __init__(self, nums):
        self.sums = [0] * (len(nums) + 1)
        for i, val in enumerate(nums):
            self.update(i, val)

    def update(self, i, val):
        i += 1
        while i < len(self.sums):
            self.sums[i] += val
            i += i & -i

    def prefix_sum(self, i):
        i += 1
        res = 0
        while i > 0:
            res += self.sums[i]
            i -= i & -i          # get parent
        return res

    def range_sum(self, i, j):
        return self.prefix_sum(j) - self.prefix_sum(i-1)

```