Product of Array Except self

nums = [1,2,3,4]

Brute Force

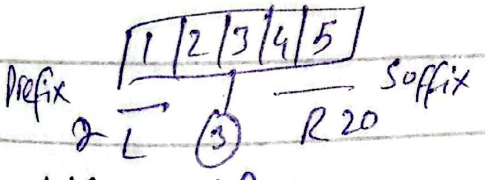
ans = [24,12,8,6]

Overall product = 24

ans = [0,1,2,3]
[24,12,8,6] by dividing the overall product with each indexth value

```
for(i=0; i<n; i++){
    prod=1
    for(j= 0; j<n; j++){
        if(i!=j){
            prod *= num[j]
        }
    ans[i] = prod
}
```

Optimal Approach



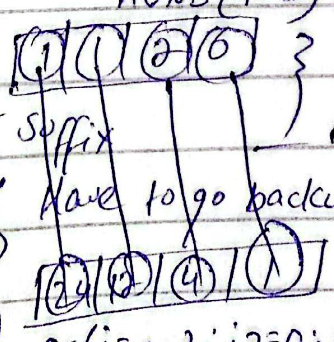
ANS = $\underline{L*R}$
2*20
40

L.S

For Prefix of All elements

```
for(i=1;i<n; i++){
    Prefix[i]= prefix[i] *
            nums[i-1]
}
```

O(n)



For Suffix

O(n) (Have to go backward)

```
for(i=n-2; i>=0; i--){
    suff[i]= soff[i+1] *
            nums[i+1];
}
```

O(n) { Ans = Suff * pref

we do'nt need extra spaces for prefixes and suffixes.

We can optimize space complexity as well.

```
for(int i=1; i<n; i++){
  ans[i] = ans[i-1] * nums[i-1];
}

int suffix = 1;
 for(int i=n-2; i>=0; i--){
  suffix *= nums[i+1];
   ans[i] *= suffix;
 }
 return ans;
}
```