b'

Google Interview Onsite (University Grad \xe2\x80\x93 2020)

- Difficulty Level :\nMedium
- Last Updated :\n21 Apr, 2020

Question : Given an array having 2n elements you can choose n elements from either end of the array such that the values obtained result in maximum sum.

Examples:

```
Input : 1 3 100 25 20 4  \r\nOutput : 103 \r\n
```

Approach: Initially, I tried recursive approach by showing both possibilities of an element that it can either be included or excluded, but he told to optimize it and I came up with prefix sum approach. **Idea: The main idea behind the prefix sum approach was if we select \xe2\x80\x98x\xe2\x80\x99 elements from left we can select \xe2\x80\x98n-x\xe2\x80\x99 elements from the right.**

```
int function(int arr[])
{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0Int n = arr.size();
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0Int lpref[n], rpref[n]; // for left and right prefix sum
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0Lpref[0] = arr[0], rpref[n - 1] = arr[n - 1];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0For(int i = 1; i < n; i++)
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0Lpref[i] = Lpref[i - 1] + arr[i];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0For(int i = n - 2; i >= 0; i--)
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0{
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0rpref[i] = rpref[i + 1] + arr[i];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0Int maxm = INT_MIN, m = n / 2;
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0For(int i = 0; i < m - 1; i++)
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0{
maxm =max(maxm, lpref[i]+rpref[n-1-i];
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0}
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0maxm = max(maxm, lpref[m - 1]);
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0maxm = max(maxm, rpref[n - m]);
\xc2\xa0\xc2\xa0\xc2\xa0\xc2\xa0return maxm;
}
```

My Personal Notes\n*arrow_drop_up*

Add your personal notes her

Save

'