

Maximum Contiguous Sum

- ❖ กำหนดให้: A คือลำดับของจำนวน $\langle a_1, a_2, \dots, a_n \rangle$
- ❖ จงหา : ผลรวมของช่วงของข้อมูลที่ติดกันใน A ที่ผลรวมมีค่ามากที่สุด
- ❖ ตัวอย่าง : A = $\langle 31, -41, 59, 26, -53, 58, 97, -93, -23, 84 \rangle$

187

MCS : Brute Force

❖ ลองทุกช่วง: มีกี่ช่วง ?

- ❖ ช่วงที่ยาว 1, 2, 3, ..., n
มีจำนวน $n, n-1, n-2, \dots, 1$ ตามลำดับ
- ❖ มีทั้งหมด $n+(n-1)+\dots+1 = \Theta(n^2)$ ช่วง

```
MCS( A[1..n] ) {  
    max = 0  
    for (i = 1 to n)  
        for (j = i to n) {  
            sum = 0  
            for (k = i to j) sum += A[k]  
            if (sum > max) max = sum  
        }  
    return max  
}
```

$\Theta(n^3)$

สามารถปรับปรุงให้หาผลรวมทุกช่วงได้ใน $\Theta(n^2)$

MCS : Brute Force

$$\sum a[i..j+1] = \sum a[i..j] + a[j+1]$$

```
MCS( A[1..n] ) {  
    max = 0  
    for (i = 1 to n) {  
        sum = 0  
        for (j = i to n) {  
            sum = 0  
            sum += A[j]  
            if (sum > max) max = sum  
        }  
    }  
    return max  
}
```


$\Theta(n^2)$

MCS : Divide & Conquer

- ❖ แบ่งเป็นชุดซ้าย ชุดขวา
- ❖ หา MCS ชุดซ้าย : S_L , หา MCS ชุดขวา : S_R
- ❖ หา MCS ข้ามจากซ้ายไปขวา : S_M
- ❖ คำตอบคือ $\max(S_L, S_R, S_M)$
- ❖ 31, -41, 59, 26, -53, 58, 97, -93, -23, 84
- ❖ 31, -41, **59, 26**, -53
- ❖ **58, 97**, -93, -23, 84
- ❖ 31, -41, **59, 26, -53, 58, 97**, -93, -23, 84
- ❖ max. sum = $\max(85, 155, 187) = 187$

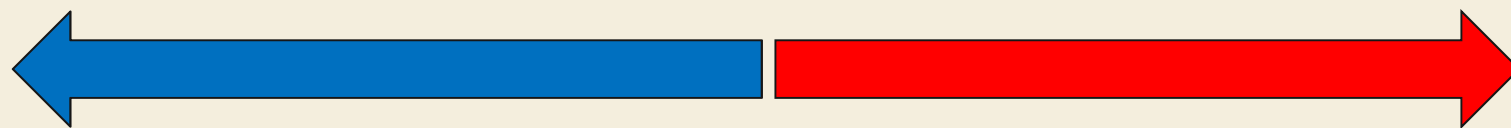
MCS : Divide and Conquer

```
MCS( A[1..n] ) {  
    return MCS_DQ( A[1..n] )  
}  
  
MCS_DQ( A[left .. right] ) {  
    if ( left > right ) return 0  
    mid = (left + right)/2  
    SL = MCS_DQ( A[left..mid] )  
    SR = MCS_DQ( A[mid+1..right] )  
    SM = MCS_LR( A[left..right] )  
    return max( SL, SR, SM )  
}
```

$$t(n) = 2t(n/2) + f(n)$$


MCS : การหาผลรวมมากที่สุดข้ามฝั่ง

31, -41, 59, 26, -53, 58, 97, -93, -23, 84



22, -9, 32, -27, -53, 58, 155, 62, 39, 123

187

MCS : Divide and Conquer

```
MCS_LR( A[left..right] ) {  
    mid = (left + right)/2  
    maxL = 0; sumL = 0  
    for ( i = mid downto left )  
        sumL += A[i]  
        if (sumL > maxL) maxL = sumL  
  
    maxR = 0; sumR = 0  
    for ( i = mid+1 to right )  
        sumR += A[i]  
        if (sumR > maxR) maxR = sumR  
  
    return maxL + maxR  
}
```

$\Theta(n)$

MCS : Divide and Conquer

```
MCS( A[1..n] ) {  
    return MCS_DQ( A[1..n] )  
}  
  
MCS_DQ( A[left .. right] ) {  
    if ( left == right ) return A[left]  
    mid = (left + right)/2  
    SL = MCS_DQ( A[left..mid] )  
    SR = MCS_DQ( A[mid+1..right] )  
    SM = MCS_LR( A[left..right] )  
    return max( SL, SR, SM )  
}
```



$$t(n) = 2t(n/2) + \Theta(n)$$

$$\Theta(n \log n)$$

MCS : อีกวิธี $\Theta(n)$

31, -41, 59, 26, -53, 58, 97, -93, -23, 84

0 31 -41 59 85 32 90 187 94 71 155

```
MCS( A[1..n] ) {  
    max = 0, sum = 0  
    for (i = 1 to n)  
        sum += A[i]  
        if (sum > max) max = sum  
        if (sum < 0) sum = 0  
    }  
    return max  
}
```

