Competitive Algorithm Design and Practice
Maximum Sub-array Sum
2014/03/19

Yi Long, Lu (mike199250)

mike199250@gmail.com

http://myweb.ncku.edu.tw/~f74991073/20140319_DP.zip

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan

NCKU CSIE Programming Contest Training Course

Made By mike199250

Maximum Sub-array Sum

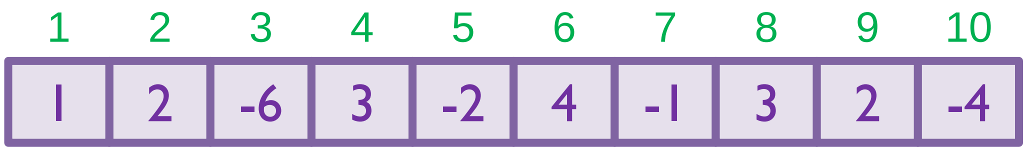
Made By mike199250

MSS

- Find a sub-array which contains continuous elements and the summation is maximum.

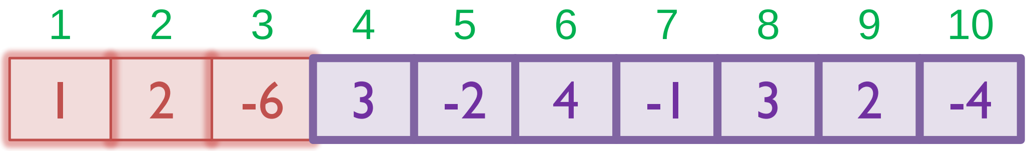
1	2	3	4	5	6	7	8	9	10
1	2	-6	3	-2	4	-1	3	2	-4

MSS



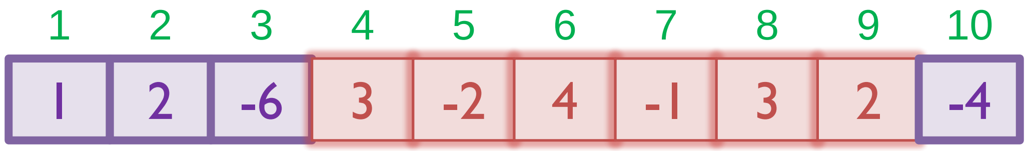
1	2	3	4	5	6	7	8	9	10
1	2	-6	3	-2	4	-1	3	2	-4

- `sum: -3`



1	2	3	4	5	6	7	8	9	10
1	2	-6	3	-2	4	-1	3	2	-4

- `Sum: 9`



1	2	3	4	5	6	7	8	9	10
1	2	-6	3	-2	4	-1	3	2	-4

MSS

- Naïve solutions:
  - For every sub-array, check if its summation is maximum.

- Time-complexity:
  - Every sub-array, O(N^2)
  - For each array, summation needs O(N)
  - Total: O(N^2*N) => O(N^3)

Made By mike199250

MSS

- Build prefix-sum in O(N)
- sum[i]=sum[i-1]+num[i]

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
sum	1	3	-3	0	-2	2	1	4	6	-2

- summation(i,j) = sum[j]-sum[i-1]
- Summation(4,9) = sum[9]-sum[3] = 6 - (-3)
  = 9

MSS

- Better solutions:
  - For every sub-array, check if its summation is maximum.

- Time-complexity:
  - Every sub-array, $O(N^2)$
  - For each array, summation needs $O(1)$
  - Total: $O(N^2*1)$ => $O(N^2)$

MSS

- Even Better solutions:
  - Divide & Conquer

- Time-complexity:
  - Total: O(NlogN)

MSS

- Every array must have a right end.
- Let`s say MSS[k] is the maximum summation of sub-array that ended at index k.
- if MSS[k-1]>0 then MSS[k] will be

  MSS[k-1] + num[k].
- if MSS[k-1]<0 then MSS[k] will be

  num[k].

Made By mike199250

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	?	?	?	?	1					

Made By mike199250

MSS

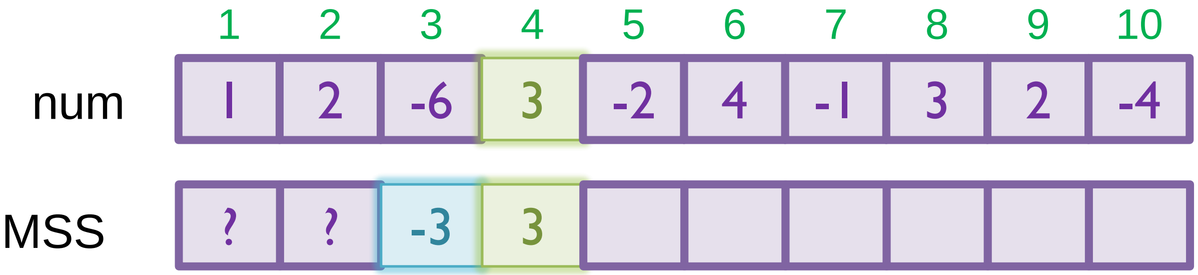
	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	?	?	?	?	1	5				

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	?	?	-3							

Made By mike199250

MSS



	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	?	?	-3	3						

MSS

- What do we want to know?
  - Maximum summation ended at k
  - i.e. MSS[k]
- How can we get that?
  - Find the previous number with longest LIS.
  - MSS[k] = {

$$\begin{array}{ll} \text{num[k]+MSS[k-1]}, & \text{MSS[k-1]} \geq 0 \\ \text{num[k]} & , \text{MSS[k-1]} < 0 \end{array}$$

}

or MSS[k] = max(0,MSS[k-1])+num[k]

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS										

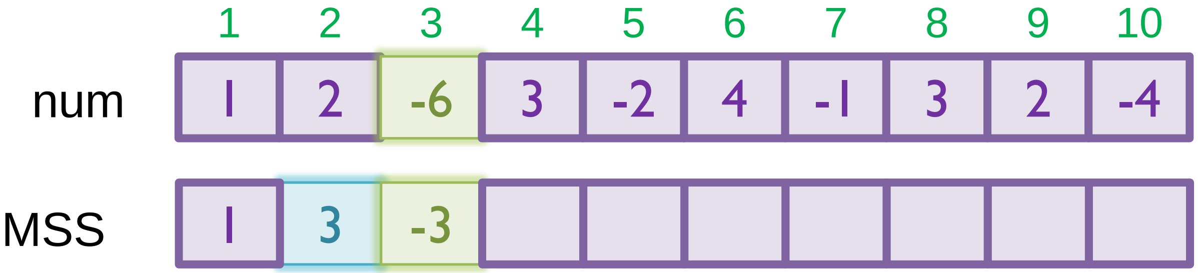
MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1									

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3								

MSS



	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3							

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3							

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3						

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1					

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5				

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5	4			

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5	4	7		

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5	4	7	9	

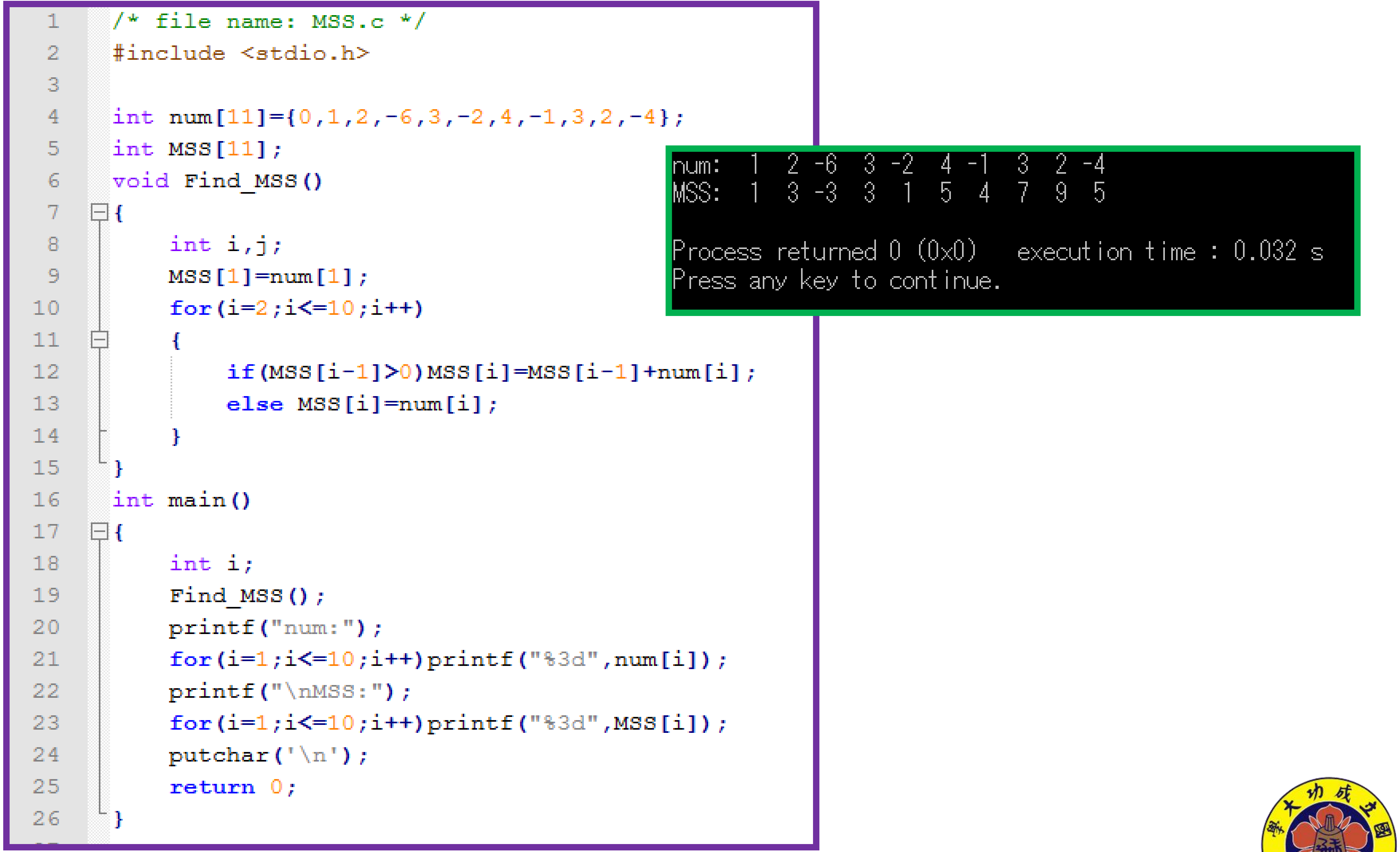
MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5	4	7	9	5

MSS

	1	2	3	4	5	6	7	8	9	10
num	1	2	-6	3	-2	4	-1	3	2	-4
MSS	1	3	-3	3	1	5	4	7	9	5

MSS



```c
/* file name: MSS.c */
#include <stdio.h>

int num[11]={0,1,2,-6,3,-2,4,-1,3,2,-4};
int MSS[11];
void Find_MSS()
{
    int i,j;
    MSS[1]=num[1];
    for(i=2;i<=10;i++)
    {
        if(MSS[i-1]>0)MSS[i]=MSS[i-1]+num[i];
        else MSS[i]=num[i];
    }
}
int main()
{
    int i;
    Find_MSS();
    printf("num:");
    for(i=1;i<=10;i++)printf("%3d",num[i]);
    printf("\nMSS:");
    for(i=1;i<=10;i++)printf("%3d",MSS[i]);
    putchar('\n');
    return 0;
}
```

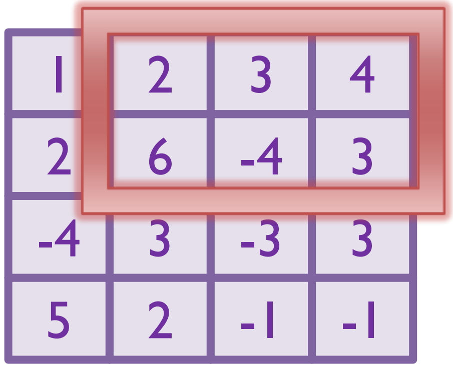
```
num:  1  2 -6  3 -2  4 -1  3  2 -4
MSS:  1  3 -3  3  1  5  4  7  9  5

Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```

Uva 10684

Made By mike199250

Learn more!

- How about two dimension?



1	2	3	4
2	6	-4	3
-4	3	-3	3
5	2	-1	-1

- Summation is 14

Learn more!

- For each sub-array…… O(N^4) because we need to determine up, down, left, right.

- Try to slice them into many 1D array.

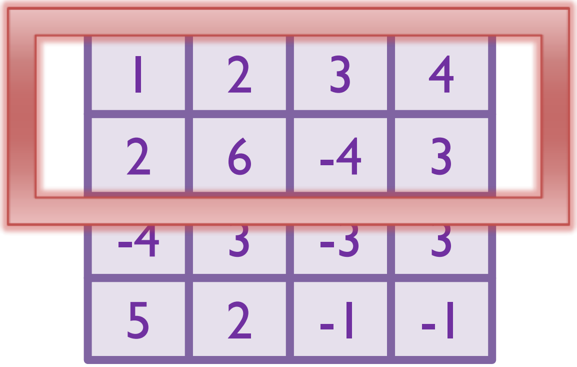
Learn more!

1	2	3	4

1	2	3	4
2	6	-4	3
-4	3	-3	3
5	2	-1	-1

Learn more!

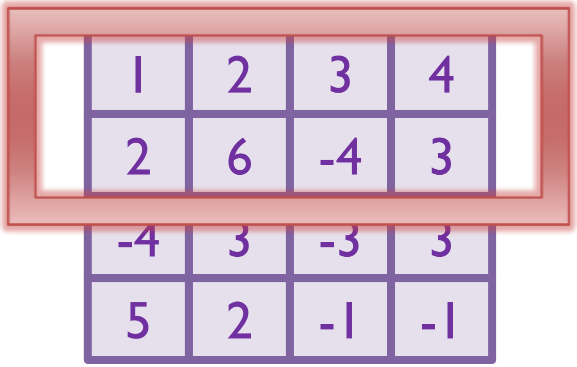
1	2	3	4
+	+	+	+
2	6	-4	3



1	2	3	4
2	6	-4	3
-4	3	-3	3
5	2	-1	-1

Learn more!

3	8	-1	7

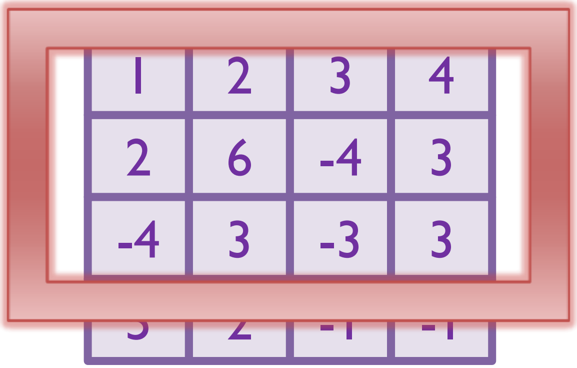


1	2	3	4
2	6	-4	3
-4	3	-3	3
5	2	-1	-1

Made By mike199250

Learn more!

-1	11	-4	10



1	2	3	4
2	6	-4	3
-4	3	-3	3
3	2	-1	-1

Learn more!

1	5	-4	2

1	2	3	4
2	6	4	3
-4	3	-3	3
5	2	-1	-1