Maximum Sum

Given a 2-dimensional (Max 100x100) array of positive and negative integers, find the sub-rectangle with the largest sum. The sum of a rectangle is the sum of all the elements in that rectangle. In this problem the sub-rectangle with the largest sum is referred to as the maximal sub-rectangle. A sub-rectangle is any contiguous sub-array of size 1x1 or greater located within the whole array.

An Example

The maximal sub-rectangle of the array:

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	8	0	-2

is in the lower-left-hand corner:

9	2
-4	1
-1	8

and has the sum of 15.

Brute Force Method

```
static int BruteForce(int x, int y){
  int max=0;int s=0;
  for(int si=1;si<=x;si++){
    for(int sj=1;sj<=y;sj++){
      for(int ei=si;ei<=x;ei++){
        for(int ej=sj;ej<=y;ej++{
          for(int ci=si;ci<=ei;ci++){
            for(int cj=sj;cj<=ej;cj++){
              s+=square[ci][cj];
            }
          }
          if(s>max)
             max=s;
        }
      }
    }
  }
  return max;
}
```

Worst case time: $100^6 = 1000000 * 10^6$, Not doable!

Dynamic Programming

How to save time from repeated computation?

First, get a new 2-dimensional array, each entry (x,y) is the sum of the sub-square (0,0,x,y) in the given array:

```
for(int i=1; i<s; i++){
  for(int j=1; j<s; j++){
    sum[i][j]=array[i][j]+sum[i][j-1]
              +sum[i-1][j]-sum[i-1][j-1];
  }
}
```

Then, to compute the sum of the sub-square (sx,sy,ex,ey), simply compute:

```
sum[ex][ey]-sum[sx-1][ey]
  -sum[ex][sy-1]+sum[sx-1][sy-1]
```

Why?

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	8	0	-2

Modified Program

```
static int BruteForce(int x, int y){
  int max=0;int s=0;
  for(int si=1;si<=x;si++){
    for(int sj=1;sj<=y;sj++){
      for(int ei=si;ei<=x;ei++){
        for(int ej=sj;ej<=y;ej++{
          if((s=sum[ei][ej]-sum[si-1][ej]
              -sum[ei][sj-1]+sum[si-1][sj-1])>max)
             max=s;
        }
      }
    }
  }
  return max;
}
```

Eliminate 2 for loops! Worst case time: $100^4 = 100 * 10^6$, Finish in a few seconds!