Assignment 2

Thomas Vinther & Jens Kristian Refsgaard Nielsen

12-09-18

Given an $n \times n$ array we wish to find the rectangular sub matrix with the property that the sum of its entries is bigger than the sum of any other rectangular sub matrix. Consider the following pseudo code
Let A be the input array
Let R and S be $n \times n \times n$ arrays and K be an $n \times n \times n \times n$ array, with all entries equal 0

```
for i = [1..n]
    for j = [1..n]
        for k = [j..n]
            for l = [j..k]
                R[i,j,k] += A[i,l]
                S[i,j,k] += A[l,j]
```

After this piece of pseudocode we have stored all the row sums of various lengths in the R array, and likewise we have the column sums in the S array in $O(n^4)$ time. Consider next

```
for i = [1..n]
    for j = [1..n]
        for k = [i+1..n]
            for l = [j+1..k]
                for h = [i..k]
                    K[i,j,k,l] += R[h,j,l]
```

Now the K array contains the sums of all rectangles with top left cornor in A[i,j] and bottom right cornor in A[k,l], this unfortunately takes $O(n^5)$ time. Now we are prepared to compare.

```
maxSoFar = 0
for i = [1..n]
    for j = [1..n]
        maxSoFar = max(A[i,j],maxSoFar)
        for k = [j+1..n]
            maxSoFar = max(R[i,j,k],maxSoFar)
            maxSoFar = max(S[i,j,k],maxSoFar)
            for l = [i+1..k]
                maxSoFar = max(K[i,j,k,l],maxSoFar)
```

This bit runs in $O(n^4)$ time.
We conclude that in total our algorithm runs in $O(n^4) + O(n^5) + O(n^4) = O(n^5)$ time.