Solution to the Problem 4.4: Maximal Sum Subarray

Main Idea

Given an array $A[1..n]$, we need to find, for each $1 \le i \le n$, the maximum sum of a subarray that covers the position i.

Assume that a subarray $A[l..r]$ has the maximum sum among those that cover the position i. Its sum is equal to $A[l] + A[l+1] + \cdots + A[r] = \sum_{j=l}^{r} A[j]$. At the same time, this sum can also be computed as the sum of the elements of the whole array minus its prefix sum and minus its suffix sum:

$$\sum_{j=l}^{r} A[j] = \sum_{j=1}^{n} A[j] - \sum_{j=1}^{l-1} A[j] - \sum_{j=r+1}^{n} A[j].$$

Since we want the sum of the subarray to be as large as possible, we need to make the corresponding prefix and suffix sums as small as possible. Thus, for each i, we need to be ably to find quickly the smallest prefix sum on $A[1..i-1]$ and the smallest suffix sum on $A[i+1..n]$. For this, we first compute all prefix and suffix sums, we then find the corresponding minimums. Overall, this gives an $O(n)$ algorithm.

Implementation Details

Let $\mathit{prefixSum}[k] = \sum_{i=1}^{k} A[k]$. Then, $\mathit{prefixSum}[k]$ for all $1 \le k \le n$ can be computed by a single scan of the array A:

```
$\mathit{prefixSum}[1] \leftarrow A[1]$
for $k$ from 2 to $n$:
  $\mathit{prefixSum}[k] \leftarrow \mathit{prefixSum}[k-1] + A[k]$
```

An array *suffixSum* defined by $\mathit{suffixSum} = \sum_{i=k}^{n} A[i]$ can be computed in a similar fashion (but by scanning A from right to left).

Let now $\mathit{minPrefixSum}[k] = \min\{\mathit{prefixSum}[i] \colon 1 \le i \le k\}$. This array can be also computed in time $O(n)$:

```
$\mathit{minPrefixSum}[1] \leftarrow \mathit{prefixSum}[1]$
for $k$ from 2 to $n$:
  $\mathit{minPrefixSum}[k] \leftarrow \min(\mathit{minPrefixSum}[k-1], \mathit{prefixSum}[k])$
```

An array *minSuffixSum* defined by $\mathit{minSuffixSum} = \min\{\mathit{suffixSum}[i] \colon k \le i \le n\}$ is computed similarly.

Finally, let $\mathit{sum} = \sum_{i=1}^{n} A[i]$ (can be computed in time $O(n)$). Then, the maximum sum of a subarray covering a position $1 < i < n$ is equal to

$$\mathit{sum} - \mathit{minPrefixSum}[i-1] - \mathit{minSuffixSum}[i+1]$$

The cases $i = 1$ and $i = n$ are treated similarly.