Introduction to Analysis of Algorithms — January 20, 2022
Boston University — CS 330, Spring 2022

Homework 1

Due Wednesday, January 26th at 11:59 pm ET on Gradescope

Problem 1. *(5 points)*

Algorithm 1: Mystery(A, k)

Input: A is an array of integers, indexed 0 through n-1. k is a positive integer.

1 n = length(A);
2 $S = zeros(n-k)$ `/* length n-k array of 0s */`
3 **for** $j = 0$ *to* $k-1$ **do**
4     $S[0] = S[0] + A[j]$;
5 **for** $i = 1$ *to* $n-k-1$ **do**
6     $S[i] = S[i-1] - A[i-1] + A[i+k-1]$;
7 **return** S ;

1. Explain in English what Mystery is computing in terms of A, i.e. explain how the value contained in $S[i]$ relate to the values stored in A for every $i = 0, 1, \dots n-k-1$.
2. Compute the number of numerical operations (additions and subtractions) that are performed in this algorithm. Give an *exact* formula as a function of n and k. Explain your computation in a few words.
3. Formally prove the relationship between $S[i]$ and A that you stated.

   (*Hint:* State a clear loop invariant for lines 5–6, and explain why it holds for each $S[i]$.)

Problem 2. *(5 points)*

Consider the following pseudocode:

Algorithm 2: TestAlg(A)

Input: A is an array of real numbers, indexed from 0 to $n-1$

$n = length(A)$;
for $j = 1$ *to* $\lfloor \frac{n-1}{2} \rfloor$ **do**
    $k = n - j$;
    $A[j] = A[j] + A[k]$;
    $A[k] = A[j] - A[k]$;
    $A[j] = A[j] - A[k]$;

Which of the following statements are true at the end of every iteration of the **for** loop? State for each whether it is true or false.

For the ones that you select as always true, prove why this is the case. (Hint: for each index j explain what value is stored in $A[j]$ at the end of iteration j.)

1. The sub-array $A[1 \ldots j]$ contains its original contents in their original order.
2. The sub-array $A[1 \ldots j]$ contains the original contents of sub-array $A[(n-j) \ldots n-1]$ in reverse order.
3. The sub-array $A[1 \ldots j]$ contains its original contents in reverse order.
4. The sub-array $A[(n-j) \ldots n-1]$ contains its original contents in their original order.
5. The sub-array $A[(n-j) \ldots n-1]$ contains the original contents of sub-array $A[1 \ldots j]$ in reverse order.
6. The sub-array $A[(n-j) \ldots n-1]$ contains its original contents in reverse order.

Problem 3. *(10 points)*

Consider the following $SwapAlg(\,)$ algorithm.

Algorithm 3: SwapAlg(A)

```
Input: A is an array of n positive integers indexed 0 to n − 1.
n = length(A) ;
swaps = 0;
for i = 0 to n − 2 do
    for j = 0 to n − i do
        if A[j] > A[j + 1] then
            A[j] = A[j] + A[j + 1];
            A[j + 1] = A[j] − A[j + 1];
            A[j] = A[j] − A[j + 1];
            swap + +;
Output: A, swap
```

1. In what order are the values initially in A if $SwapAlg$ executes the smallest possible number of swaps? In what order are the values initially to result in the largest number of swaps?(Explain in English)
2. Compute the exact number of swaps that are performed on the worst case input as a function of n. Explain your formula.

3. Consider the *decAlg*( ) algorithm.

Algorithm 4: decAlg(A)

Input: A is an array of n positive integers indexed 0 to $n-1$.

```
n = length(A) ;
dec = 0;
for i = 1 to n do
    j = i − 1;
    while j > 1 AND A[j − 1] > A[j] do
        temp = A[j − 1];
        A[j − 1] = A[j];
        A[j] = temp;
        j − −;
        dec + +;
```

Output: A, dec

4. Observe that the array returned by $swapAlg(A)$ and $decAlg(A)$ are the same. Prove that the values $swap$ and dec are equal. (Hint: Prove the following statement: for any two indices x and y the values initially stored in $A[x]$ and $A[y]$ are swapped in $swapAlg$ if and only if they are swapped in $decAlg$.)