CS300: Homework #1

Due on September 27, 2016 at 10:30am

Prof. Sunghee Choi

20160051 Ohjun Kwon

Problem 1

Part A

$T(n) = T(n-1) + \Theta(n)$

Part B

We can use unrolling method to solve this recurrence as below.

$$\begin{aligned}
T(n) &= T(n-1) + cn \\
&= T(n-2) + c(n-1) + cn \\
&\vdots \\
&= \sum_{i=1}^{n} ci \\
&= c\sum_{i=1}^{n} i \\
&= c\frac{n(n-1)}{2} \\
&= c_2 n^2 + c_1 n + c_0 \in \Theta(n^2)
\end{aligned}$$

where c_2, c_1, c_0 are constants. □

Problem 2

Part A

In order to be A' is called a sorted permutation of A, we need to show that A' is made of the initial array A, and it is put in order that satisfies $\forall i, j \;$ s.t. $1 \le i \le j \le n, A[i] \le A[j]$ on the termination.

Part B

Loop invariant At the beginning of each iteration of the loop, the element $A[j]$ is the smallest element in the subarray $A[j...n]$ where $n = A.length$. Also, these values are originated from the initial subarray $A[j...n]$.

Initialization The loop starts with $j = n$. The subarray $A[j...n] = A[n...n]$ only contains one element, which is apparently the smallest value in the subarray.

Maintenance The smallest element in the subarray $A[j...n]$ is $A[j]$ by the loop invariant stated above. Line 4 in the *BUBBLESORT* is executed only if the element $A[j-1]$ is bigger than $A[j]$. This extends our range $j...n$ that loop invariant condition($A[j-1]$ is the smallest element in the subarray $A[j-1...n]$, and $A[j-1...n]$ is a permutation of the previous sequence) holds to $j-1...n$, which maintains the loop invariant on the next loop.

Termination The loop terminates when the variable j reaches i, which makes $A[i]$ minimum in the subarray $A[i...n]$ and the subarray $A[i...n]$ is a permutation of the subarray $A[i...n]$ at the initial point.

Part C

Problem 2 continued on next page. . .

Loop invariant At the beginning of each iteration of the loop, the subarray $A[1...i-1]$ is sorted array that contains $i-1$ smallest elements of the original array $A[1...n]$. In addition, the subarray $A[i...n]$ contains the remaining values waiting to be sorted, i.e. $A[1...n]$ is a permutation of the previous values.

Initialization The loop starts with $i = 1$. The subarray $A[1...0]$ is an empty array, which is considered as sorted(is a permutation and formed a sorted sequence).

Maintenance At the beginning of the loop, it begins with $A[1...i-1]$ sorted, which contains $i-1$ smallest values. After the execution of the 2-4th line in *BUBBLESORT*, the smallest element in the subarray $A[i...n]$ will be located at $A[i]$, and the new subarray $A[i...n]$ is a permutation of the previous $A[i...n]$ as we proved at **Problem 2 Part B**. As $A[i]$ is the i-th smallest element in $A[1...n]$, the $A[1...i]$ is new subarray that satisfies the loop invariant.

Termination The loop terminates when the variable i reaches n, and the loop invariant says that $A[1...n-1]$ contains $n-1$ smallest elements in increasing order, in other words, sorted. This ensures the last value $A[n]$ is the biggest value in the array $A[1...n]$, which also satisfies the ordering, and all elements are from the original array.

Part D

The worst case for the bubblesort is the "backward sorted sequence". It makes the comparison and exchange happens everytime, which is executed $n-1$, $n-2$, $\cdots$, 1 times. (Total $\frac{n(n-1)}{2}$ times). Therefore, $\Theta(n^2)$ is the worst-case time complexity, which is same with insertion sort.

Problem 3

Prove: For any two functions $f(n)$ and $g(n)$, we have $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

Solution

($\Rightarrow$)
$f(n) = \Theta(g(n))$ means that $f(n)$ satisfies the statement 1.

$$\exists c_1, c_2, n_0 > 0 \;\text{ s.t. } \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \tag{1}$$

In addition, the statement 1 can be broken into two following statements.

$$\exists c_2, n_0 > 0 \;\text{ s.t. } \forall n \geq n_0, 0 \leq f(n) \leq c_2 g(n) \tag{2}$$
$$\exists c_1, n_0 > 0 \;\text{ s.t. } \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \tag{3}$$

The statement 2 is the definition of the $f(n) = O(g(n))$, and the statement 3 is the definition of the $f(n) = \Omega(g(n))$. These conclusions were derived from the given statement (statement 1). □

($\Leftarrow$)
This way also can be proved by doing the proof exactly opposite way. $f(n) = O(g(n))$ gives us the statement 2, and $f(n) = \Omega(g(n))$ gives us the statement 3.
Next, if we combine those two statements into one, it looks like the statement 1, which means $f(n) = \Theta(g(n))$. □

Problem 4

Part A

$T(n) = 2T(n/4) + \sqrt{n}$

Solution

$a = 2$, $b = 4 \Rightarrow n^{\log_b a} = \sqrt{n}$; $f(n) = \sqrt{n}$

Case 2 on the master method: $f(n) = \Theta(\sqrt{n}\mathrm{lg}^0 n)$, $k = 0$

$\therefore T(n) = \Theta(\sqrt{n}\mathrm{lg}n)$

Part B

$T(n) = T(n-2) + n^2$

Solution

$$\begin{aligned}
T(n) &= n^2 + T(n-2) \\
&= n^2 + (n-2)^2 + T(n-4) \\
&= \sum_{i=0}^{n/2} (n-2i)^2 \\
&= \sum_{i=0}^{n/2} (n^2 - 4ni + 4i^2) \\
&= \frac{n^3}{2} - 4n\frac{\frac{n}{2}\left(\frac{n}{2}-1\right)}{2} + 4\frac{\frac{n}{2}\left(\frac{n}{2}+1\right)\left(2\frac{n}{2}+1\right)}{6} \\
&\in \Theta(n^3)
\end{aligned}$$

$\therefore T(n) = \Theta(n^3)$

Part C

$T(n) = 7T(n/3) + n^2$

Solution

$a = 7$, $b = 3 \Rightarrow n^{\log_b a} = n^{\log_3 7}$; $f(n) = n^2$

Case 3 on the master method: $f(n) = \Omega(n^{log_3 7+\epsilon})$, for some $\epsilon > 0$ ($\because log_3 7 < 2$)

$7(n/3)^2 \leq cn^2$ for $c = 7/9$

$\therefore T(n) = \Theta(n^2)$

Part D

$T(n) = 2T(n/2) + 3T(n/3) + n^2$

Solution