Recitation Problems - Com S 311

Week of Feb 5^{th} - Feb 10^{th}

1. Derive the worst-case runtime of the following loop structure as a function of n and determine its Big-O upper bound. You must show the derivation of the end result. Assume the runtime is expressed as follows:

$$\text{Runtime } = \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=i}^{j} 1$$

$$\begin{aligned}
\text{Runtime } &= \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=i}^{j} 1 \\
&= \sum_{i=1}^{n}\sum_{j=i}^{n}(j-i+1) \\
&= \sum_{i=1}^{n}\frac{(n-i+1)(n-i+2)}{2} \qquad \text{[See equation (1) below]} \\
&= \frac{1}{2}\sum_{i=1}^{n}(n^2-ni+2n-ni+i^2-2i+n-i+2) \\
&= \frac{1}{2}\sum_{i=1}^{n}(n^2-2ni-3i+3n+2+i^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}(n^2-i(2n+3)+3n+2+i^2) \\
&= \frac{1}{2}\left[n^3-\frac{n(n+1)}{2}(2n+3)+3n^2+2n+\frac{n(n+1)(2n+1)}{6}\right] \\
&= \frac{1}{2}\left[n^3-\frac{2n^3+5n^2+3n}{2}+3n^2+2n+\frac{2n^3+3n^2+n}{6}\right] \\
&= \frac{1}{2}\left[\frac{6n^3-6n^3-15n^2-9n+18n^2+12n+2n^3+3n^2+n}{6}\right] \\
&= \frac{1}{12}\left[2n^3+6n^2+4n\right] \\
&\in O(n^3)
\end{aligned}$$

$$\begin{aligned}
\sum_{j=i}^{n}(j-i+1) &= 1+2+3+...+(n-i+1) \\
&= \frac{(n-i+1)(n-i+2)}{2}
\end{aligned} \tag{1}$$

2. You are given an array A of integers, sorted in increasing order, and a target integer T. Write an algorithm to verify whether there exist **two** integers x and y in A such that $x+y=T$. Discuss the derivation of the runtime of your algorithm.

   Example: $A = [3,6,9,10,11]$

   - $T = 16 \quad \rightarrow \quad$ Answer = Yes
   - $T = 14 \quad \rightarrow \quad$ Answer = Yes
   - $T = 11 \quad \rightarrow \quad$ Answer = No
   - $T = 22 \quad \rightarrow \quad$ Answer = No

Algorithm 1 Solution

```
i = 0;    // leftmost index of array
j = n − 1;    // rightmost index of array
found = false; // boolean flag
while i < j && !found do
    if A[i] + A[j] < T then // implies that we need to look for bigger number on left
        i + +;
    else if A[i] + A[j] > T then // implies that we need to look for smaller number on right
        j − −;
    else
        found = true;
return found;
```

Runtime Analysis:

The runtime depends on the number of times the while loop iterates in the worst case. The loop iterates as long as $j - i > 0$. Initially, $j - i$ is $n - 1$. In each iteration, this value decrements by 1. Therefore, the number of times the loop iterates is bounded by n. In the body of the loop, we do constant time operations. Hence, the runtime $\in O(n)$.