Question 3

(a)

The function recursiveFnc(int[] A, int j, int k) is recursively called on lines 5 and 6. With $n = k - j + 1$ entries. We will determine the input size given to each recursive call.

Note that since $n = k - j + 1$, we have $k = n + j - 1$, and $j = k - n + 1$.

On line 5, recursiveFnc() is called with input size:

$$\lfloor \frac{j+k+1}{2} \rfloor - 1 - \lfloor \frac{3j+k+1}{4} \rfloor + 1$$

$$= \lfloor \frac{n+2j}{2} \rfloor - \lfloor \frac{n+4j}{4} \rfloor, \text{ since } k = n + j - 1$$

$$= \lfloor \frac{n}{2} \rfloor + j - \lfloor \frac{n}{4} \rfloor - j$$

$$= \lfloor \frac{n}{2} \rfloor - \lfloor \frac{n}{4} \rfloor$$

On line 6, recursiveFnc() is called with input size:

$$k - \lfloor \frac{j+3k+3}{4} \rfloor + 1$$

$$= k - \lfloor \frac{4k+4-n}{4} \rfloor + 1, \text{ since } j = k - n + 1$$

$$= k + \lceil \frac{-4k-4+n}{4} \rceil + 1$$

$$= k + \lceil \frac{n}{4} \rceil - k - 1 + 1$$

$$= \lceil \frac{n}{4} \rceil$$

By the condition on line 3, we get a base case when $j \geq k - 6$, or $j \geq n + j - 1 - 6 \implies n \leq 7$. But, since 7 in this base case does not depend on n, we will approximate it by 1. Furthermore, for every recursive call of recursiveFnc() the function localFnc() is called with cost $n^{1/2}$ when acting on input size of n entries. Altogether, we derive the recurrence relation

$$T(n) = \begin{cases} 1, & n = 1 \\ T(\lfloor \frac{n}{2} \rfloor - \lfloor \frac{n}{4} \rfloor) + T(\lceil \frac{n}{4} \rceil) + n^{1/2}, & n > 1 \end{cases}$$

Note that we do not add 1 from the cost of the condition on line 3, since it does not depend on n.

(b)

Assume that $n = 4^r$ for some $r \geq 0$. Then $r = 0$ gives $n = 1$; the base case, and $r \geq 1$ gives $n \geq 4$; the recursive case. In the recursive case, since $n = 4^r$, it must be true that $4|n$, and thus $2|n$. Therefore $\frac{n}{2}$ and $\frac{n}{4}$ are both integers. Thus, we can simplify our recursive case to

$$T(\lfloor \frac{n}{2} \rfloor - \lfloor \frac{n}{4} \rfloor) + T(\lceil \frac{n}{4} \rceil) + n^{1/2}$$

$$= T(\frac{n}{2} - \frac{n}{4}) + T(\frac{n}{4}) + n^{1/2}$$

$$= T(\frac{n}{4}) + T(\frac{n}{4}) + n^{1/2}$$

$$= 2T(\frac{n}{4}) + n^{1/2}$$

So, we can simplify the recursive relation to be

$$T(n) = \begin{cases} 1, & n = 1 \\ 2T(\frac{n}{4}) + n^{1/2}, & n > 1 \end{cases}$$

(c)

Inputs:

n		$n^{1/2}$		
$n/4$	$(\frac{n}{4})^{1/2}$		$(\frac{n}{4})^{1/2}$	
$n/4^2$	$(\frac{n}{4^2})^{1/2}$	$(\frac{n}{4^2})^{1/2}$	$(\frac{n}{4^2})^{1/2}$	$(\frac{n}{4^2})^{1/2}$

The sum/cost of row 1 is $n^{1/2}$.

The sum/cost of row 2 is $2(\frac{n}{4})^{1/2} = (\frac{2n^{1/2}}{4^{1/2}}) = n^{1/2}$

The sum/cost of row 3 is $4(\frac{n}{4^2})^{1/2} = (\frac{4n^{1/2}}{(4^2)^{1/2}}) = n^{1/2}$

All paths are identical, where the input size is divided by 4 each time until we reach 1, where we reach a base case. So the length of any given branch is $\log_4 n$.

Then, since each row costs $n^{1/2}$, we calculate the total cost to be $T(n) = n^{1/2} \log_4 n$. So, we conjecture that $T(n) \in \Theta(n^{1/2} \log(n))$.

(d)

Recall the recurrence relation from part (b). Let $a = 2, b = 4, c = 1, d = 1$, and $f(n) = n^{1/2}$. Since the relation is of proper form, $a, c \geq 1, b > 1$, and $d \geq 0$, we may use the Master Theorem to determine the cost of this relation.

Note that $\log_b(a) = \log_4(2) = 1/2$. So, $n^{\log_4(2)} = n^{1/2}$. Therfore, $f(n) \in \Theta(n^{\log_4(2)})$.

From the Master Theorem, we can then determine that

$$T(n) \in \Theta(n^{1/2} \log(n))$$

Which is the same answer we calculated in (c).