DISMATH Q2

Name: KEY Section:

1. Use the Insert sort algorithm to sort alphabetically the following list.

S, P, T, W, N

Write the state of the **full list** for every assignment operation or when an element changes value during the execution of the algorithm. Use asterisk (*) to denote elements that are already sorted. (Solution in the test booklet). How many comparisons are needed in this case? 7 (19 is the actual)

What is the time complexity (big-Theta notation) of Insert sort? $\Theta(n^2)$

Simplified Count: 7

```
Enter 5 characters
S
P
T
W
N

Input Array:
{S, P, T, W, N}

Insertion sorting ...
j = 1; temp = P
{S, S, T, W, N}
{P, S, T, W, N}
j = 2; temp = T
{P, S, T, W, N}
j = 3; temp = W
{P, S, T, W, N}
j = 4; temp = N
{P, S, T, W, W}
{P, S, T, T, W}
{P, S, S, T, W}
{P, P, S, T, W}
{N, P, S, T, W}

Sorted Array:
{N, P, S, T, W}
```

Actual Count: 19

```
Input Array:
{S, P, T, W, N}

Insertion sorting ...
Entering outer for loop...
j = 1; temp = P
Entering inner for loop...
{S, S, T, W, N}
{P, S, T, W, N}
Entering outer for loop...
j = 2; Entering while loop...
Entering while loop...
temp = T
{P, S, T, W, N}
Entering outer for loop...
j = 3; Entering while loop...
Entering while loop...
Entering while loop...
temp = W
{P, S, T, W, N}
Entering outer for loop...
j = 4; temp = N
Entering inner for loop...
{P, S, T, W, W}
Entering inner for loop...
{P, S, T, T, W}
Entering inner for loop...
{P, S, S, T, W}
Entering inner for loop...
{P, P, S, T, W}
{N, P, S, T, W}
count = 19

Sorted Array:
{N, P, S, T, W}
```

2.

Let $f : \mathbb{N} \longrightarrow \mathbb{R}$ be defined by

$$f(n) = \frac{n^4 + \log_2 n}{n^2 + 1}.$$

a. Θ (__n^2_) b. Upper-bound witnesses: C =_2; $n \geq 1$_ c. Lower-bound witnesses: _C = ½ $n \geq 1$

$$\begin{aligned}
|f(n)| &= \left|\frac{n^4 + \log n}{n^2 + 1}\right|, \quad \text{so} \quad n > 0 \\
&= \frac{n^4 + \log n}{n^2 + 1}, \quad \text{all terms} \ \geq 0 \\
&\leq \frac{n^4 + n}{n^2}, \quad \log n \leq n \\
&\leq \frac{2n^4}{n^2} \quad \text{for } n \geq 1 \\
&= 2n^2 \\
|f(n)| &= \left|\frac{n^4 + \log n}{n^2 + 1}\right| \\
&= \frac{n^4 + \log n}{n^2 + 1}, \quad \text{all terms} \ \geq 0 \\
&\geq \frac{n^4}{n^2 + n^2}, \quad \text{for } n \geq 1 \\
&= \frac{1}{2}n^2 \\
\therefore \frac{1}{2}n^2 &\leq |f(n)| \leq 2n^2 \\
\therefore\ f(n) &= \Theta(n^2)
\end{aligned}$$

3. If A = {1, 2, 3}, B = {2, 4, 6, 8} and the universal set U = {1, 2, 3, ...} find
a. A – B = {_1, 3_} b. B' = {_1, 3, 5, 7, 9, 10, 11,..._}
c. Power set of A = {_{1}, {2}, {3}, {1, 2}, {2, 3}, {1,3}, {1, 2, 3}, {}}

4. Give a proof of or a counterexample to the following statement: (Solution in the test booklet)

$$A \cap (B \cup C) = (A \cup B) \cap (A \cup C)$$

A = {a}; B = {b}; C = {b};

$$\{\} \neq \{a, b\}$$

5. Construct a table showing the interchanges that occur at each step when bubble sort is applied to the following list: 6, 4, 5, 7, 3

steps	lists

Continue the solution to the booklet
if necessary.

REFER TO DISCUSSION NOTES ….

6. Given the following function:

$$f(x) = (x^2 + 5x + 3)(x + 2\log x)$$

a. f(x) is $O(x^4)$: True/ False __True____

b. f(x) is $O(x^3)$: True/ False __True_____

c. f(x) is $O(x^2)$: True/ False __False_____

d. f(x) is $O(x^3 \log x)$: True/ False ____True___

e. f(x) is $O(x^2 \log x)$: True/ False ____False____

7. Given:

Procedure **A** (n: positive integer)
$s := 0$
for $i := 1$ **to** n
  for $j := 1$ **to** i
    $s := s + j$
return s

a. Suppose that procedure A is started with input n = 4. Then what number is returned by the algorithm?
___________________20________________________

b. The worst-case time complexity of procedure A is: ________________$\Theta(n^2)$___________

8. (a) How many functions are there from {1,2} to {a,b,c}? __9__

(b) How many of these functions are one-to-one? __6__

(c) How many of these functions are onto? __0__

(d) How many of these functions are bijective? __0__

g_1	$\{(1,a), (2,a)\}$
g_2	$\{(1,a), (2,b)\}$
g_3	$\{(1,a), (2,c)\}$
g_4	$\{(1,b), (2,a)\}$
g_5	$\{(1,b), (2,b)\}$
g_6	$\{(1,b), (2,c)\}$
g_7	$\{(1,c), (2,a)\}$
g_8	$\{(1,c), (2,b)\}$
g_9	$\{(1,c), (2,c)\}$

9. Count the number of comparisons for the ff. algorithm: ______________________________

What is its time complexity? __________________________________

Require: $\{a_1, a_2, \ldots, a_i, \ldots, a_n\}_{\neq} \in \mathbb{Z}$, where $a_1 < a_2 < \ldots < a_n$; $x \in \mathbb{Z}$
Ensure: $result = k$, where $(a_k = x)$ and $k \in \{1, \ldots, n\}$ if the element is found; otherwise $k = -1$

```
i ← 1
j ← n
while i < j do
    mid ← ⌊(i + j)/2⌋
    if x > a_mid then
        i ← mid + 1
    else
        j ← mid
    end if
end while
if x == a_i then
    result ← i
else
    result ← −1
end if
```

Solution: For simplicity, assume there are $n = 2^k$ elements in the list $a_1, a_2, \ldots, a_n$, where k is a nonnegative integer. Note that $k = \log n$. (If n, the number of elements in the list, is not a power of 2, the list can be considered part of a larger list with 2^{k+1} elements, where $2^k < n < 2^{k+1}$. Here 2^{k+1} is the smallest power of 2 larger than n.)

At each stage of the algorithm, i and j, the locations of the first term and the last term of the restricted list at that stage, are compared to see whether the restricted list has more than one term. If $i < j$, a comparison is done to determine whether x is greater than the middle term of the restricted list.

At the first stage the search is restricted to a list with 2^{k-1} terms. So far, two comparisons have been used. This procedure is continued, using two comparisons at each stage to restrict the search to a list with half as many terms. In other words, two comparisons are used at the first stage of the algorithm when the list has 2^k elements, two more when the search has been reduced to a list with 2^{k-1} elements, two more when the search has been reduced to a list with 2^{k-2} elements, and so on, until two comparisons are used when the search has been reduced to a list with $2^1 = 2$ elements. Finally, when one term is left in the list, one comparison tells us that there are no additional terms left, and one more comparison is used to determine if this term is x.

Hence, at most $2k + 2 = 2\log n + 2$ comparisons are required to perform a binary search when the list being searched has 2^k elements. (If n is not a power of 2, the original list is expanded to a list with 2^{k+1} terms, where $k = \lfloor \log n \rfloor$, and the search requires at most $2\lceil \log n \rceil + 2$ comparisons.) It follows that in the worst case, binary search requires $O(\log n)$ comparisons. Note that in the worst case, $2\log n + 2$ comparisons are used by the binary search. Hence, the binary search uses $\Theta(\log n)$ comparisons in the worst case, because $2\log n + 2 = \Theta(\log n)$.

K. Rosen, Discrete Mathematics and its Applications, (7th ed.) pp. 220

10. Given a set of two-dimensional points, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, in the Cartesian plane.

a. Write a pseudocode to find the farthest pair of points by computing the distances between all pairs of the n points and determining the largest distance.

Farthest Pair

Algorithm 14 Find the farthest pair of points by computing the distances between all pairs of the n points and determining the largest distance.

```
Require: {(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)} where x_k, y_k ∈ ℝ
Ensure: result = ((x_i, y_i), (x_j, y_j)) where
  distance((x_i, y_i), (x_j, y_j)) is maximum
  max ← 0
  for i = 2 to n do
    for j = 1 to i − 1 do
      if (x_j − x_i)^2 + (y_j − y_i)^2 > max then
        max ← (x_j − x_i)^2 + (y_j − y_i)^2
        result ← ((x_i, y_i), (x_j, y_j))
      end if
    end for
  end for
```

b. Write a pseudocode to sort the points according to the abscissa of the corresponding points.

USE BUBBLE SORT or ANY OTHER SORTING ALGORITHMS

c. Write a pseudocode to sort the points according to the ordinate of the corresponding points.

USE BUBBLE SORT or ANY OTHER SORTING ALGORITHMS

d. Give the time complexity estimate (Big Theta) for each of the previous algorithms.

All are $\Theta(n^2)$.

*** END ***