1. (3 points) Give a succinct English description of what Magic computes.

Solution: Magic computes the largest difference between two values in its input list.

2. (4 points) Suppose $T(n)$ is the running time of Magic. Give a recursive definition of $T(n)$.

Solution:

$$T(1) = d_1 \quad T(2) = d_2 \quad T(3) = d_3$$

$$T(n) = 2T(n-1) + cn + p$$

3. (4 points) What is the height of the recursion tree for $T(n)$?

Solution: We hit the base case when $n - k = 3$, where $k$ is the level. So the tree has height $n - 3$.

4. (4 points) How many leaves are in the recursion tree for $T(n)$?

Solution: $2^{n-3}$
01 Frog($p_1, \ldots, p_n$ : list of $n$ 2D points, $n \geq 3$)
02 
03 if ($n = 3$)
04 
05 return the largest of $d(p_1, p_2)$, $d(p_1, p_3)$, and $d(p_2, p_3)$
04 
05 else
06 
07 $x = \text{Frog}(p_2, p_3, p_4, \ldots, p_n)$ \quad i.e. remove $p_1$
06 
07 $y = \text{Frog}(p_1, p_3, p_4, \ldots, p_n)$ \quad i.e. remove $p_2$
07 
08 $z = \text{Frog}(p_1, p_2, p_4, \ldots, p_n)$ \quad i.e. remove $p_3$
08 
08 return the largest of $x$, $y$, and $z$

The function $d(p, q)$ returns (in constant time) the straight-line distance between two points $p$ and $q$.

1. (5 points) Suppose $T(n)$ is the running time of Frog on an input array of length $n$. Give a recursive definition of $T(n)$. Assume that setting up the recursive calls in lines 5-7 takes constant time.

Solution:

$$T(3) = c$$

$$T(n) = 3T(n - 1) + d$$

2. (4 points) What is the height of the recursion tree for $T(n)$? Solution:

Solution: We hit the base case when $n - k = 3$, where $k$ is the level. So the tree has height $n - 3$.

3. (3 points) How many leaves are in the recursion tree for $T(n)$?

Solution: $3^{n-3}$

4. (3 points) What is the big-Theta running time of Frog?

Solution: $\Theta(3^n)$
procedure Reorder(a$_1$, ..., a$_n$) \[\text{input is an array of } n \text{ integers}\]

for $i := 1$ to $n - 1$

\[\text{min} := i\]

for $j := i$ to $n$

\[\text{if } a_j < a_{\text{min}} \text{ then } \text{min} := j\]

swap(a$_i$, a$_{\text{min}}$) \[\text{interchange the values at positions } i \text{ and } \text{min} \text{ in the array}\]

Swap takes constant time.

1. (4 points) If the input is 10, 5, 2, 3, 8, what are the array values after two iterations of the outer loop?

**Solution:** After one iteration, the array contains 2, 5, 10, 3, 8.
After the second iteration, it contains 2, 3, 10, 5, 8.

2. (4 points) Let $T(n)$ be the number of times that line 5 is executed. Express $T(n)$ using summation notation, directly following the structure of the code.

**Solution:** The $i$th time through the outer loop, the inner loop runs $n - i + 1$ times. So the total number of times that line 5 executes is:

\[\sum_{i=1}^{n-1} (n - i + 1)\]

3. (4 points) Find an (exact) closed form for $T(n)$. Show your work.

**Solution:** If we break apart the sum and then substitute in a new index variable $p = n - i$ we get:

\[\sum_{i=1}^{n-1} (n - i + 1) = (n - 1) + \sum_{i=1}^{n-1} (n - i) = (n - 1) + \sum_{p=1}^{n-1} p = (n - 1) + \frac{n(n-1)}{2}\]

Simplifying, we get

\[(n - 1) + \frac{n(n-1)}{2} = n - 1 + \frac{1}{2}n^2 - \frac{1}{2}n = \frac{1}{2}n^2 + \frac{1}{2}n + 1\]

4. (3 points) What is the big-theta running time of Reorder?

**Solution:** $\Theta(n^2)$
01  snape(a_1, \ldots, a_n): a list of \( n \) positive integers \\
02  if (\( n = 1 \)) return \( a_1 \) \\
03  else if (\( n = 2 \)) return \( \max(a_1, a_2) \) \\
04  else if (\( a_1 < a_n \)) \\
05    return snape(a_2, \ldots, a_n) \\
06  else 
07    return snape(a_1, \ldots, a_{n-1})

Max takes constant time. Removing the last element of a list takes \( O(n) \) time.

1. (5 points) Let \( T(n) \) be the running time of snape. Give a recursive definition of \( T(n) \).
   
   **Solution:**
   
   \[ T(1) = c \]
   \[ T(2) = d \]
   \[ T(n) = T(n - 1) + pn \]

2. (3 points) What is the height of the recursion tree for \( T(n) \)?
   
   **Solution:** We hit the base case when \( n - k = 2 \), where \( k \) is the level. So the tree has height \( n - 2 \).

3. (3 points) What is amount of work (aka sum of the values in the nodes) at level \( k \) of this tree?
   
   **Solution:** Notice that the tree doesn’t branch, so there is only one node at each level. So the total amount of work at level \( k \) is \( p(n - k) \).

4. (4 points) What is the big-theta running time of snape? Briefly justify and/or show your work?
   
   **Solution:** \( \Theta(n^2) \)

   If you aren’t sure why, notice that the sum of all the non-leaf nodes is \( \sum_{k=1}^{n-3} p(n - k) \). If we move the constant \( p \) out of the summation and substitute in the new index value \( j = n - k \), we get

   \[ p \sum_{j=3}^{n-1} j = p \sum_{j=1}^{n-1} j - 3 = p \frac{(n-1)n}{2} - 3 = \frac{p}{2} n^2 - \frac{p}{2} n - 3 \]

   The dominant term of this is proportional to \( n^2 \).