

# ESO207: Data Structures and Algorithms

## Programming Assignment 1

Due Date: 1st February, 2021

Total Number of Pages: 2

Total Points 240

\* Submit all the graph, recurrence relation and code on Moodle

### Question 1. Triangle Migration

The faraway mythical land of Artemis, inhabited by Dryads, is in the shape of a giant equilateral triangle directed "upwards". Dryads are isolationist creatures and they live alone in equilateral triangles. Dryads also have another amusing feature. After one year, every Dryad in Artemis replicates into 4 dryads and relocates to a smaller equilateral triangle directed either "upwards" or "downwards". Each year, when a dryad divides into four smaller dryads: three smaller dryad shift to smaller triangles directed in the same direction as their parent dryad, and one small dryad shifts to a smaller triangle directed in the opposite direction. Then each year this process repeats. The figure below illustrates this process.

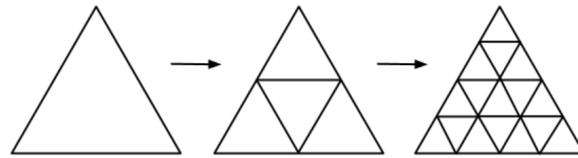


Figure 1: How Dryads form their localities each year

Help the original dryad find out how many smaller dryads have localities that point "upwards" in  $n$  years. As the number can be quite large, you should print it modulo  $1000000007(10^9 + 7)$ .

#### Example :

Let  $n = 2$ . Output will be 10.

#### Explanation :

Artemis is the first equilateral triangle with 1 dryad. At  $n=1$ , it divides into 4 dryad with 3 having smaller localities pointing "upwards" and 1 directed "downwards". At  $n=2$ , each of these 4 replicate, thus giving the answer.

- (a) (10 points) Let  $U(n)$  and  $D(n)$  be the total possible localities directed "upwards" and "downwards" respectively. Give the recurrence relations for  $U(n)$  and  $D(n)$ .  $U(1) = 3, D(1) = 1$ .
- (b) (40 points) **Using recursive method**, write a program that output  $U(n), D(n)$  for a given  $n$ .  
 $1 \leq n \leq 10^{15}$
- (c) (60 points) **Using iterative method**, write a program that output  $U(n), D(n)$  for a given  $n$ .  
 $1 \leq n \leq 10^{15}$

Name:

Rollno:

- 
- (d) (100 points) **Using matrix exponentiation method**, write a program that output  $U(n)$  for a given  $n$ .  
 $1 \leq n \leq 10^{15}$

- (e) (30 points) Generate a graph (**Input (n) vs. Time required (micro-seconds)**) for each of the above methods. **Each graph consist of 10 marks**. You can use the python script in the following link for graph generation :

<https://gist.github.com/akashks1998/283410b3f014d5fc68b4766e32b35552>

For measuring time you can use other library functions. Refer to <https://www.geeksforgeeks.org/how-to-measure-time-taken-by-a-program-in-c/> for more details on measuring time in C.

**Inputs (n values) for which you have to generate Input vs Time required graph. Scale of the required time should be in micro-seconds.**

- Recursive method:  $n = 1, 2, 3, \dots, 20$ .
- Iterative method:  $n = 1, 500, 1000, 1500, \dots, 10000$ .
- Power method:  $n = 1, 10, 100, 1000, \dots, 10^{15}$ . [Use logarithmic scale for representing  $n$  values on X-axis in power method, i.e, values in X-axis should be from 0, 1, 2, ..., 15]  
For each input in recursive and iterative method, calculate  $U(n), D(n)$  10 times and take the average time for plotting the graph. For power method the run time is quite small and will vary. So, for each input in this method, calculate  $U(n)$  1000 times before taking the average time for graph generation.

**Note :**

- Parts **b, c and d** of Question 1 have to be answered through a contest in Hackerrank. The contest has 3 challenges, each corresponding to a part. You have to submit your code through the contest. Following is the link to the contest: <https://www.hackerrank.com/eso207a-pa1-2020-21-ii>
- For the parts **b, c and d**, you will not be granted any marks if you submit your code/solution by any method other than what is expected through the question in the challenges (irrespective of the number of test cases that pass for your code). For example, if you submit power methods solution for the recursive code, you will get 0 for that part.
- Don't use memorisation for the **recursive method implementation**.
- Your codes will be checked for possible plagiarism of any sorts. If we find such cases, then we will possibly award an F grade.
- Allowed Languages for challenge code submission : C, C++
- Allowed libraries : stdio.h for C and iostream for C++
- You need to submit a total of 4 files - 3 corresponding to the 3 programs (through Hackerrank), and 1 file containing solution of **parts (a) and (e) and your Hackerrank username**.