

TD 1 :

1 Basic mathematics

Question 1 Among the following expressions, what are the ones that are $O(\text{some polynomial in } n)$?
(here, k is a fixed constant, i.e., $k = O(1)$)

- | | | | |
|-------------------------------|-----------------------------|-----------------|----------------------|
| 1. $2^{5 \log n}$ | 4. n^π | 7. n^{2025} | 10. $n^k * 2^k$ |
| 2. $n^2 + 6n + \frac{1}{2^n}$ | 5. $\frac{n}{\log \log n}$ | 8. n^n | 11. $k^2 + k \log k$ |
| 3. $2^{\sqrt{n}}$ | 6. $\log^6 n (\log \log n)$ | 9. $n^{\log n}$ | 12. 7 |

Question 2 — What is the number of subsets of a set with n elements ?

- What is the number of sequences with n elements (each between 0 and n) so that each element appears exactly one ?
- What is the number of numbers with 10 digits (each between 0 and 9) such that there are exactly two “4” and at least one “5” ?
- What is the order of magnitude of $\sum_{i=0}^n (i^2 \sum_{j=0}^i j)$?

Question 3 What is the time complexity of the following two algorithms ?

Algorithm 1 Example 1

```

1: For  $i = 1$  to  $n$  do :
2:     For  $j = 1$  to  $\ell$  do :
3:         For  $k = 1$  to  $n$  do : Print “Hello”

```

Algorithm 2 Example 2 : Fib

Require: $n \in \mathbb{N}$

```

1: If  $n \leq 1$  Return  $n$ 
2: Else Return  $Fib(n - 1) + Fib(n - 2)$ .

```

2 Graph Algorithms and Complexity

Question 4 What is the **class** P ? the **class** NP ? Give an intuitive explanation of what means to be **NP-complete**.

Question 5 For each of the following problems : say if it is known to be in P , give its formal definition (input + output) and name an algorithm to solve it and give its time-complexity.

1. **Sort** n integers ;
2. Compute a **Minimum Spanning tree** in a weighted graph ;
3. Compute a **shortest path** between two vertices in a weighted graph ;
4. Compute a **maximum matching** in a graph ;
5. **Load balancing** problem ;
6. Compute a minimum **vertex cover** in a graph ;
7. Decide if a graph is **Hamiltonian** ;
8. Decide if a graph admits an **Eulerian cycle** ;

Question 6 Give an algorithm that decide if a n -node graph is **connected**. Give its time-complexity.

Question 7 A set of vertices of a graph is **stable** if its elements are pairwise non-adjacent. Give an algorithm that computes a maximum stable set in an n -node graph. Give its time-complexity.

Question 8 A graph is **bipartite** if its vertex-set can be partitioned into two stable sets.
Prove that a graph is bipartite if and only if it does not contain a cycle with odd length.