

**Overview**

<b>1</b>	<b>Ordering Order of Growth</b>	<b>1</b>
<b>2</b>	<b>Presorting</b>	<b>1</b>
<b>3</b>	<b>Computing the range</b>	<b>2</b>
<b>4</b>	<b>Matrix Determinant</b>	<b>3</b>
<b>5</b>	<b>Leader Selection</b>	<b>4</b>

## 1 Ordering Order of Growth

List the following functions according to their order of growth from the lowest to the highest:

$$3^{2n} \quad , \quad (n-2)! \quad , \quad 5\lg(n+100)^{10} \quad , \quad 0.001n^4 + 3n^3 + 1 \quad , \quad \ln^2 n \quad , \quad 3^n \quad , \quad \sqrt[5]{n}$$

## 2 Presorting

One can check whether all elements of an array are distinct by a two-part algorithm based on the array's presorting. (Think about it.)

- a. If the presorting is done by an algorithm with a time efficiency in  $\Theta(n \log n)$ , what will be the time-efficiency class of the entire algorithm? (Explain briefly.)
  
  
  
  
  
  
  
  
  
  
- b. If the sorting algorithm used for presorting needs an extra array of size  $n$ , what will be the space-efficiency class of the entire algorithm? (Explain briefly.)

### 3 Computing the range

The **range** of a finite nonempty set of  $n$  real numbers  $S$  is defined as the difference between the largest and smallest elements of  $S$ . For each representation of  $S$  given below, describe in English an algorithm to compute the range. Indicate the time efficiency classes of these algorithms using the most appropriate notation ( $\mathcal{O}$ ,  $\Theta$ , or  $\Omega$ ).

- a. An unsorted array
- b. A sorted array
- c. A sorted singly linked list
- d. A binary search tree

## 4 Matrix Determinant

The determinant of an  $n \times n$  matrix,

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \dots & a_{n-1,n-1} \end{bmatrix}$$

denoted  $\det(A)$ , can be defined as  $a_{0,0}$  for  $n = 1$  and, for  $n > 1$ , by the recursive formula

$$\det(A) = \sum_{j=0}^{n-1} s_j a_{0,j} \det(A_j)$$

where  $s_j$  is  $+1$  if  $j$  is even and  $-1$  if  $j$  is odd;  $a_{0,j}$  is the element in row 0 and column  $j$ ; and  $A_j$  is the  $(n-1) \times (n-1)$  matrix obtained from matrix  $A$  by deleting its row 0 and column  $j$ .

- a. Setup a recurrence relation for the number of multiplications made by the algorithm implementing this recursive definition.
- b. Without solving the recurrence, what can you say about the solution's order of growth as compared to  $n!$ ?

## 5 Leader Selection

The country of Inductia has an interesting method of selecting their leader: The leader is chosen from a potential group of candidates as being the person who is known by everyone else in the group, but themselves does not know anyone else. Your task is to help select the new leader by asking individuals in the group (only) questions of the form “Do you know *<point-to-one-other-person>*?” Design an efficient algorithm to help determine, from a group of  $n$  people, who should be the leader, or determine that none of the candidates fits the selection requirement. How many questions (give an exact formula, if possible) must be asked according to your algorithm, in the worst case?