

Solucione los problemas propuestos usando el lenguaje de su preferencia (`python`, `java`, `c`) de ser posible entregue todas las soluciones en un mismo archivo, podría crear una clase `solution` y ahí definir los métodos propuestos. Si toma esta opción, su clase y métodos **NO** pueden tener otro nombre. La otra opción es que solucione los problemas usando funciones, este caso también tenga en cuenta que el nombre de las funciones no puede ser diferente al propuesto.

1. Escriba un algoritmo para el método Strassen's para multiplicación de matrices con entradas enteras. El nombre de la función/método debe ser `strassen_matrix_mult` de aridad 2. **Debe tener complejidad** $\mathcal{O}(n^{\log(7)})$, explique sin entrar en muchos detalles porque tiene esta complejidad. Se sugiere mostrar cual es la ecuación de recurrencia y explicar brevemente porque esta ecuación lleva a la complejidad deseada.
2. Impelente un algoritmo, usando el método de dividir y conquistar, que reciba un arreglo de numero enteros (`array`) y un numero entero (`n`). El algoritmo debe devolver el `n`-esimo elemento mas grande del arreglo. En otras palabras si se toma el arreglo ordenado el resultado sera el `n`-esimo elemento mas grande del mismo, no importa si hay repetidos. El nombre de la función/método debe ser `largest_nth_element` de aridad 2. **Debe tener complejidad** $\mathcal{O}(n)$.
 - `largest_nth_element([-2,2,-3,10,5,4,2,5],1) = 10`
 - `largest_nth_element([-2,2,-3,10,5,4,2,5],3) = 5`
 - `largest_nth_element([-2,2,-3,10,5,4,2,5],9) = 10`
 - `largest_nth_element([-2,2,-3,10,5,4,2,5],-1) = -2`
3. Una cadena `string` es bonita si, por cada letra del alfabeto que contiene `string`, aparece tanto en mayúsculas como en minúsculas. Por ejemplo, "abABB" es bonita porque aparecen 'A' y 'a', y aparecen 'B' y 'b'. Sin embargo, "abA" no lo es porque aparece 'b', pero no 'B'. Dada una cadena `string`, devuelve la subcadena más larga de `string` que sea bonita. Si hay varias, devuelve la subcadena de la primera aparición. Si no hay ninguna, devuelve una cadena vacía. Impelente un algoritmo, usando el método de dividir y conquistar, que solucione el problema con una complejidad **Debe tener complejidad** $\mathcal{O}(n \log n)$. El nombre de la función/método debe ser `nice_string` de aridad 1.