**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA Escuela Profesional de Ingeniería de Sistemas**

**DISEÑO Y ANALISIS DE ALGORITMOS Tarea de práctica – Semana 5**

**Alumno: Torres Tineo, Cristhian Anthony**

**Código: 22200050**

**Indicaciones:**

* 1. Para cada pregunta, conteste.
  2. Al final para entregar, subir al **Teams,** enviar antes de la fecha indicada, o a lo sumo hasta la fecha y hora indicada.
  3. La presente tarea de práctica es de evaluación continua de nota máxima 16 (dieciséis), evaluará:

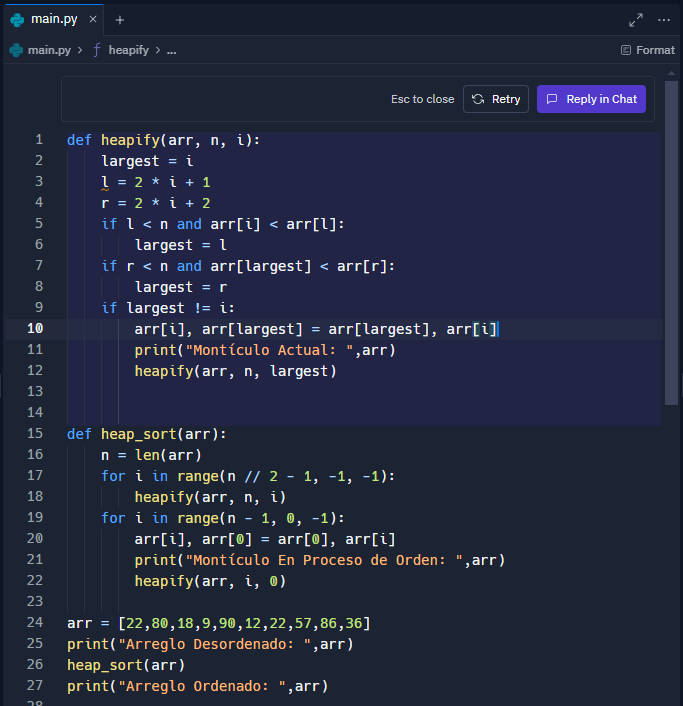
De lista 2: sorteo dado por los delegados.

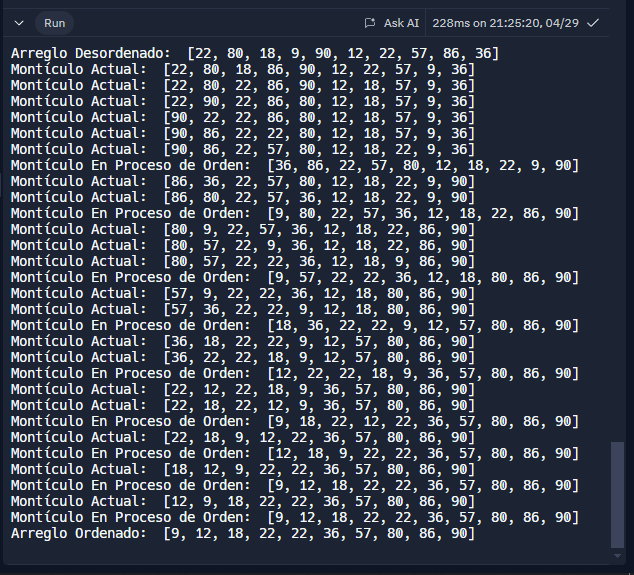
* 1. Para resolver en grupo o individual, el envío es individual, coloque su nombre y código de alumno en cada hoja.
  2. La participación en Programación Competitiva, en concurso(s) de programación podrá reforzar los conceptos, ejercicios del curso.
  3. Recuerde practicar: honestidad, responsabilidad, puntualidad.

Resuelva tres ejercicios de la Lista 3 de ejercicios – Algoritmos de ordenación (**Lista de análisis y diseño de algoritmos – Parte 02, Heaps**), implemente. (8 puntos)

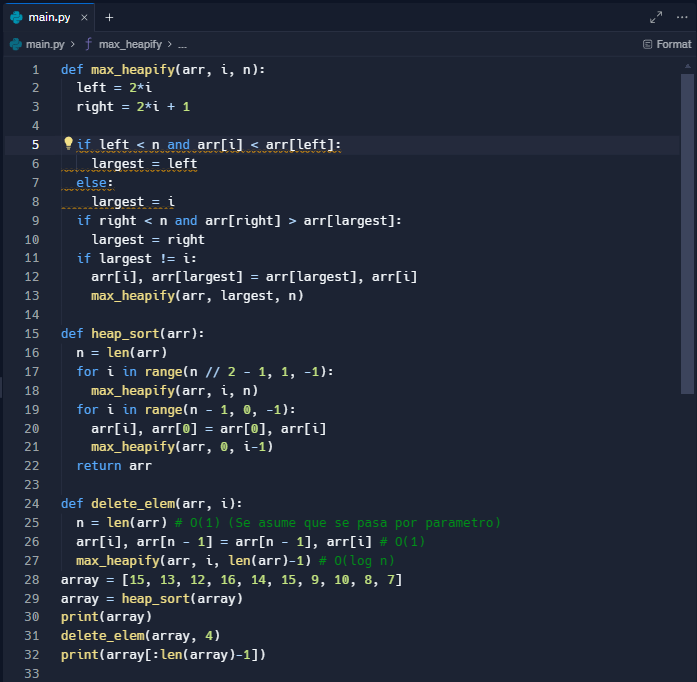
1. Ordena la siguiente secuencia usando Heap Sort

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | 80 | 18 | 9 | 90 | 12 | 22 | 57 | 86 | 36 |



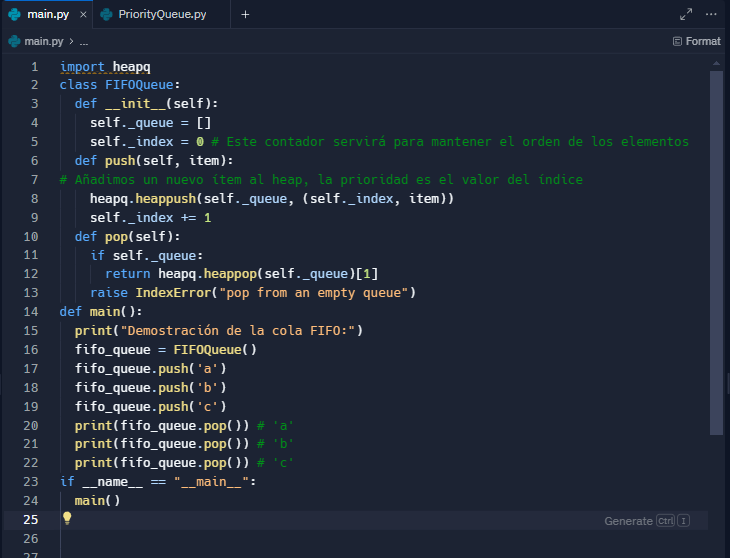


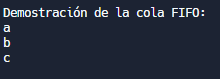
1. Describe un algoritmo que elimina un elemento con índice i de un montón máximo de n elementos. El algoritmo debe ejecutarse en tiempo O(log n).





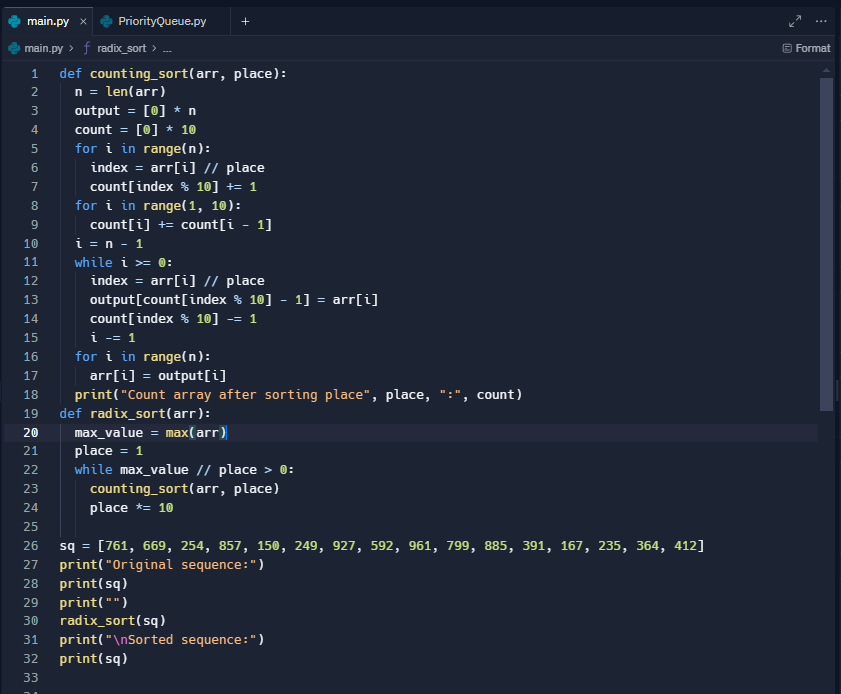
1. Muestre cómo implementar una cola FIFO con una cola de prioridad. Muestre cómo implementar una pila con una cola de prioridad.

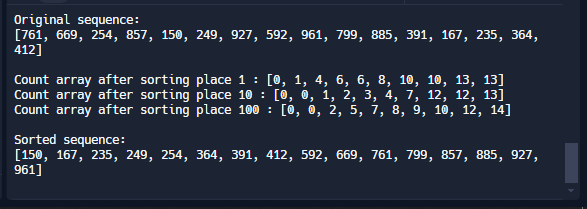




Resuelva tres ejercicios de la Lista 3 de ejercicios – Algoritmos de ordenación (**Lista de análisis y diseño de algoritmos – Parte 03, Lineal Time Sorting**), implemente. (8 puntos)

1. Radix Sort





1. Suponga que tenemos un conjunto de n registros de datos para ordenar y que la clave de cada registro tiene el valor 0 o 1.

Un algoritmo para ordenar dicho conjunto de registros podría poseer algún subconjunto de las siguientes tres

características deseables:

a) Proporcione un algoritmo que cumpla con los criterios 1 y 2.

b) Proporcione un algoritmo que cumpla con los criterios 1 y 3.

c) Proporcione un algoritmo que cumpla con los criterios 2 y 3.

d) ¿Puede usar alguno de sus algoritmos de ordenación de las partes (a)-(c) como el método de ordenación utilizada en la ordenación por radix, de modo que la ordenación por radix ordene n registros con claves de b bits en tiempo O(bn)? Explique cómo o por qué no.

Primero creamos una clase Record que representará un registro de datos con una clave binaria:

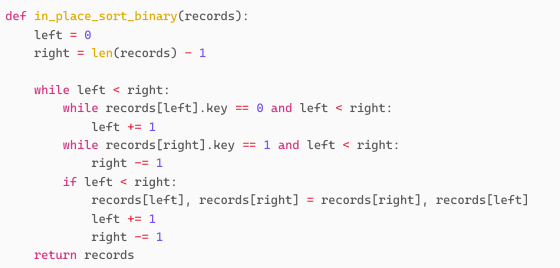


a)



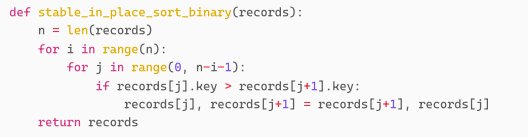
Ejecución:

b)



Ejecución:

c)



Ejecución:

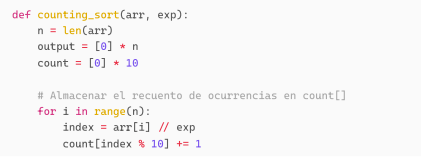
d)

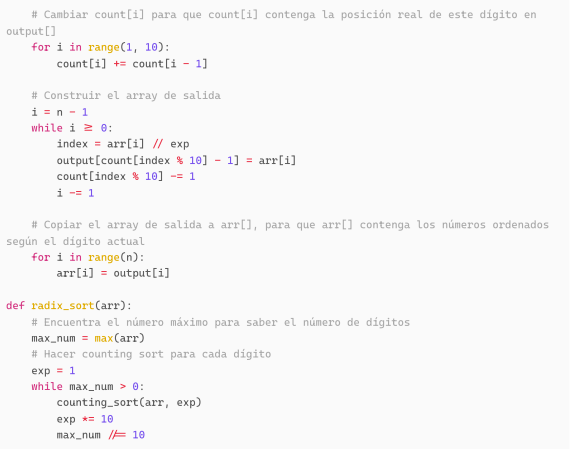


Ejecución:



1. Se le da una matriz de enteros, donde diferentes enteros pueden tener diferentes números de dígitos, pero el número total de dígitos en todos los enteros de la matriz es n. Muestre cómo ordenar la matriz en O(n) tiempo.







Ejecución



**“y juró por el que vive por los siglos de los siglos, que creó el cielo y las cosas que están en él, y la tierra y las cosas que están en ella, y el mar y las cosas que están en él, que el tiempo no sería más,”**

**Apoc. 10:6 (RVR, 1960 )**