

UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

Curso: Algoritmos y Estructuras de Datos-CC182

Evaluación: Práctica Calificada 2

Docente: Walter Cueva Chávez

Sección: CC31

Duración: 110 minutos

Inicio: 19:00 A.M.

Consideraciones

- Sus respuestas de los ejercicios deben desarrollarlo en c++.
- El envío tiene que ser enviado como archivo con extensión “.cpp” como archivo adjunto en el enlace del aula virtual.
- Todo intento de plagio parcial o total es penalizado con una calificación de cero (0).
- Cada ejercicio debe tener un caso de prueba.
- Puede usar el código implementado en clase.
- **Solamente implemente los métodos solicitados.**

Caso de Estudio

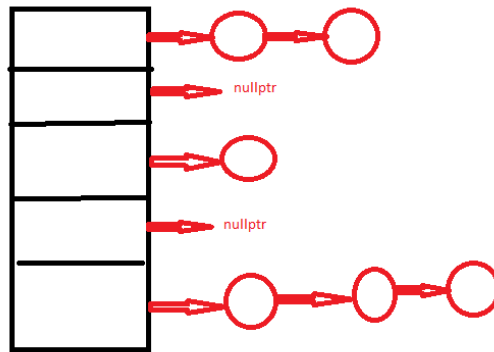
Se tiene un archivo “diccionario.csv” adjunto y se requiere que realice indexación y búsqueda entre las estructuras de árbol y hash table.

El archivo solamente contiene 2 campos nombre y significado, por ejemplo.

Nombre	Significado
A	The first letter of the English and of many other alphabets.
Aam	A Dutch and German measure of liquids- varying in different cities.
Aard-vark	An edentate mammal.
Aard-wolf	A carnivorous quadruped (Proteles Lalandii)- of South Africa- resembling the fox and hyena. See Proteles.
Aaronic	Alt. of Aaronical
Aaronical	Pertaining to Aaron- the first high priest of the Jews.

- (1 punto) Implementar una clase Registro con los atributos indicados.
- (2 punto) Implementar una clase Diccionario que realice la lectura de datos y además implemente los siguientes métodos:
 - (2 puntos) Indexar el archivo en árboles por el nombre del Nombre, use un lambda como criterio.
 - (3 puntos) muestre la cantidad de nombres duplicados en el árbol por cada letra inicial.
 - (3 puntos) Del árbol creado realice la inserción en una lista según postorden.
 - (3 puntos) Ordenar los registros por el nombre de manera descendente mediante el algoritmo de mergesort, el criterio de comparación debe ser por un puntero a función.

- (4 puntos) Implementar un HashTable que sea un arreglo de listas, en donde las colisiones se agreguen en un nodo al centro de la lista. Puede hacer uso de STL



- (2 puntos) Del HashTable anterior, indexar los registros, Ud elija la función hash considerando el menor criterio de colisión.