

Práctica 4

1) Parte I – Algoritmos de Búsqueda

- a) Implemente el algoritmo de búsqueda lineal
 1. Utilizando For
 2. Utilizando While
 3. Explique ventajas y desventajas de cada implementación.

- b) Implemente un algoritmo de búsqueda binaria.
 1. En forma iterativa
 2. En forma recursiva
 3. Explique ventajas y desventajas de cada una.

- c) Utilizando los algoritmos implementados en a y b.
 1. Busque un numero en una lista de números con:
 - 10 números
 - 100 números
 - 1000 números
 - Nota 1: utilice la lista desordenada con búsqueda lineal
 - Nota 2: utilizando la lista ordenada con búsqueda lineal y binaria

 2. Dado el objeto Persona(nombre, edad) y una lista de personas, busque el elemento con nombre “Juancito”.
 - La lista deberá contener entre 100 y 500 personas

- d) Para cada ejercicio anterior arme una tabla para verificar tiempos de respuesta según la cantidad de elementos de la lista, si esta ordenada o no. Saque conclusiones.

Cantidad	Ordenada(si/no)	Método de búsqueda	Tiempo inicio	Tiempo Fin
10	No	Lineal	----	---

- e) Compare estos resultados con el método de búsqueda provisto por python. De que forma le parece que esta implementado? Mas parecido a secuencial o a binario?

2) Parte II - Algoritmos

- a) Dada una lista con el siguiente formato: [[equipo, partidos_ganados, partidos_empatados, partidos_perdidos],[equipo, partidos_ganados, partidos_empatados, partidos_perdidos], ..., [equipo, partidos_ganados, partidos_empatados, partidos_perdidos]] . Sabiendo que la lista no esta ordenada, realice una función que reciba la lista y el nombre de un equipo y retorne la posición en la que se encuentra. Recuerde que el puntaje es igual a $\text{partidos_ganados} * 3 + \text{partidos_empatados}$.

Aclaración: Debe utilizar un algoritmo de búsqueda y justificar su elección.

- b) Dada una lista con el siguiente formato: [[equipo, partidos_ganados, partidos_empatados, partidos_perdidos],[equipo, partidos_ganados, partidos_empatados, partidos_perdidos], ..., [equipo, partidos_ganados, partidos_empatados, partidos_perdidos]] . Sabiendo que la lista esta ordenada por equipo en orden alfabético, realice una función que reciba la lista y el nombre de un equipo y retorne la posición en la que se encuentra. Recuerde que el puntaje es igual a $\text{partidos_ganados} * 3 + \text{partidos_empatados}$.

Aclaración: Debe utilizar un algoritmo de búsqueda y justificar su elección.

- c) Dada una lista en la que cada elemento tiene un primer elemento que es una letra del abecedario y un segundo elemento que es otra sub-lista con todas las palabras que empiezan con esa letra, por ejemplo: [[“A”, [“amar”, “arbol”, “azucar”]], [“B”, [“balon”, “bebe”, “bucanero”]], ..., [“Z”, [“zanahoria”, “zapiola”, “zinc”]]] . Sabiendo que los elementos estan ordenados, realice una función llamada **existePalabra** que reciba una palabra y retorne True si la encuentra o False si no la encuentra. Utilice algoritmos de búsqueda binaria.
Puede utilizar la lista que se encuentra a continuación:

```
listaPalabras = [{"A", ["amar", "arbol", "azucar"]},
{"B", ["balon", "bebe", "bucaner"]},
{"C", ["cama", "casa", "cuna"]},
{"D", ["dedo", "diamante", "duda"]},
{"E", ["ecuacion", "elefante"]},
{"F", ["falange", "fiat", "ford"]},
{"G", ["gato", "gente", "guitarra"]},
{"H", ["habas", "hepatitis", "hora"]},
{"I", ["idea", "italia", "itunes"]},
{"J", ["jacaranda", "jeque", "juego"]},
{"K", ["kapanga"]},
{"L", ["ladron", "letra", "lugar"]},
{"M", ["mama", "mozo", "muzzarella"]},
{"N", ["naturaleza", "neptuno", "norte"]},
{"O", ["oasis", "oeste", "outlet"]},
{"P", ["papa", "pepe", "pipo"]},
{"Q", ["qatar", "queso", "quiniela"]},
{"R", ["rata", "raton", "ruta"]},
{"S", ["sapo", "serie", "sinonimo"]},
{"T", ["tata", "terere", "tilo"]},
{"U", ["upe", "utilisima", "uva"]},
{"V", ["vacio", "verde", "violeta"]},
{"W", ["walter", "wii"]},
{"X", ["xavi", "xuxa"]},
{"Y", ["yacare", "yo", "yupanqui"]},
{"Z", ["zanahoria", "zapiola", "zinc"]}]]
```