

Algoritmos de Búsqueda

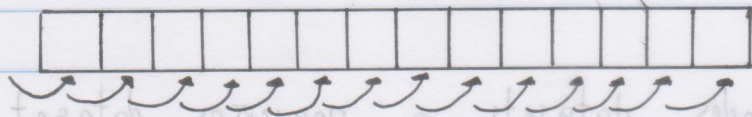
28-08-19

Búsqueda secuencial → si el arreglo es desordenado → solo se puede secuencial

→ Es el algoritmo de búsqueda más sencillo

→ Busca un elemento en una lista o un vector

→ Complejidad $O(n)$
↳ en el peor de los casos



Binary Search

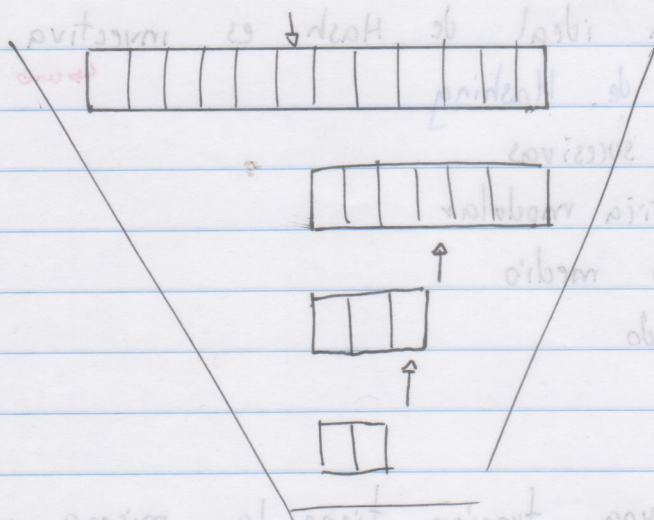
→ Busca un número en un arreglo ordenado

→ Compara el elemento buscado con el central

→ Si el central es el elemento buscado termina.

→ Si el central es menor que el buscado, se realiza la búsqueda en la mitad superior.

→ Si el central es mayor, se realiza la búsqueda en la mitad inferior



Interpolation Search

→ Modificación de Binary Search

→ El código es prácticamente el mismo, a excepción del

cálculo del elemento central

→ En cada etapa, trata de calcular donde está el elemento central.

$$\text{middle} = \text{low} + \frac{(\text{buscado} - a[\text{low}] * (\text{high} - \text{low}))}{a[\text{high}] - a[\text{low}]}$$

Hashing

- Mapea grandes datasets a pequeños dataset
- Provee una forma de buscar rápidamente. → una de las más eficientes
- Determina una función de Hash que permite buscar y encontrar un índice para una llave

Buscar un elemento en hash siempre da lo mismo
 $f(\text{llave}) \rightarrow$ índice único

- Se puede aplicar en encriptación SHA-256 / MD5... no se puede descryptar
- La forma más básica de hash es la función identidad.
- La función ideal de Hash es inyectiva → difícil encontrarlas. → uno a uno sin repeticiones.

→ Técnicas de Hashing

- Restas sucesivas
- Aritmética modular
- Cuadrado medio
- Truncado

• Folding

→ Colisiones

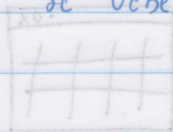
- Cuando una función tiene la misma imagen para distintas pre-imágenes
- Difícil de evitar

Checksum → Integridad archivos



Pathfinding

- Buscar un camino entre 2 puntos
 - Factores como el terreno se debe considerar
 - Ruta más corta.



- Desarrollo de juegos

- Algoritmos comunes

→ A* Actualidad se basa en dijkstra

→ Dijkstra

→ Breath-First Recorrido grafo en anchura

- Técnicas comunes "Tontas"

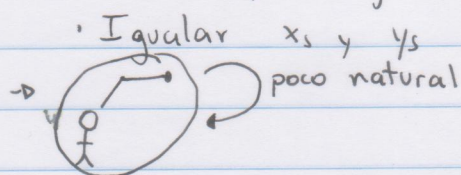
→ Breadcrumbs: El personaje deja pistas a los MPCs

→ Walltracing: Técnica de exploración "Seguir la pared izquierda"

→ Waypoint: Caminos precalculados para moverse entre nodos

- Enfoque básico

→ mover el personaje desde su origen hasta el destino



- Al usar un algoritmo de línea/vista se hace un movimiento más natural

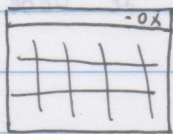
• Bresenham

→ No considera obstáculos

Si hay pocos obstáculos → Random movement

Tarea Extraclase 4

- 1) Usando matrices representando un plano visualmente



- 2) Representa el estado de la matriz en la UI

Ej: 1 en matriz coloca pelota en UI

- 3) Poder calcular basic pathfinding

- 4) Implementar un algoritmo de linea vista.

- 5) Random bouncing al encontrar obstaculos.

Sitios

- Hacker errand Rank
- Hacker noom