



# **Trabajo en proceso de revisión para el Quinto Congreso de Estudiantes de Ingeniería Informática de la Universidad de La Laguna**

**4 y 5 de Diciembre de 2019**

**Escuela Superior de Ingeniería y Tecnología de la  
Universidad de La Laguna**



PUBLISHED BY COMITÉ ORGANIZADOR DEL V CONGRESO DE ESTUDIANTES DE INGENIERÍA  
INFORMÁTICA DE LA UNIVERSIDAD DE LA LAGUNA  
BORRADOR

CESINF.WEBS.ULL.ES

*4 de Diciembre de 2019*



## Índice general

I

### Contribuciones: en proceso de evaluación

Jump search / *Anónimo*

7





# Contribuciones: en proceso de evaluación



# CESINF 2019 - Artículo en revisión

## JUMP SEARCH

Anónimo  
*ULL*

**Tutor:** Gara Miranda Valladares y Coromoto León Hernández  
*Profesoras ULL*

### Objetivo

El objetivo es entender el concepto de búsqueda en la informática. Para ello, se ha querido explicar dicho concepto mediante la siguiente técnica de búsqueda: 'Jump Search' [1].

Se ha escogido esta técnica puesto que es sencilla a la hora de buscar un elemento en un array. Además, resulta un método divertido y didáctico.

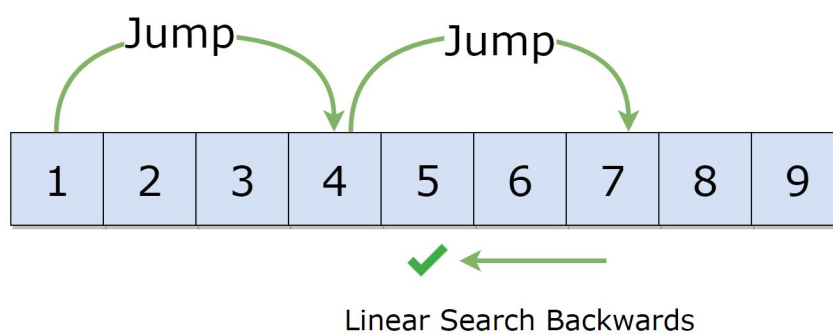


Figura 1: Jump Search [1]

### Introducción al Jump Search

Jump Search es un algoritmo de búsqueda para arrays ordenados. La idea básica del algoritmo es comprobar un menor número de elementos que si se hiciera de forma lineal.

El funcionamiento del algoritmo es el siguiente:

- Saltando un número  $n$  de elementos, se comprueba si el elemento en dicha posición es el buscado. Si es menor, se vuelve a saltar el mismo número de elementos.

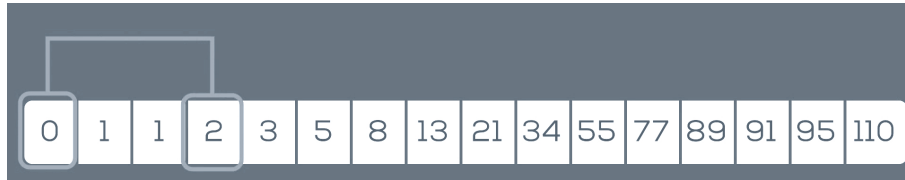


Figura 2: Búsqueda del elemento 77 - Iteración 1



Figura 3: Búsqueda del elemento 77 - Iteración 2

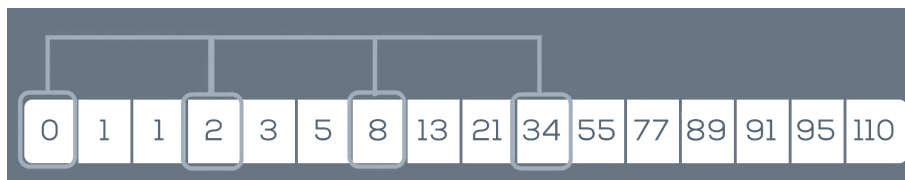


Figura 4: Búsqueda del elemento 77 - Iteración 3





Figura 5: Búsqueda del elemento 77 - Iteración 4

- Si el elemento es mayor, se empieza a recorrer el array de forma lineal retrocediendo posiciones. En caso de retroceder  $n$  posiciones o encontrar un elemento menor, el número buscado no se encuentra en el array y termina la búsqueda.
- Si el elemento se encuentra en un salto o retrocediendo posiciones, la búsqueda finaliza.



Figura 6: Encontrado el elemento 77

La eficiencia del algoritmo es  $O(\sqrt{n})$  gracias a los saltos entre elementos que se producen.

Hay diferentes métodos para determinar el número de saltos a realizar. El más eficiente es la raíz cuadrada del tamaño del array. En esta implementación y debido al público al que está dirigido se dividirá el array dependiendo del número de niños que se encuentren realizando la actividad. Si son más de 20, el número de saltos será 5, si son más de 30, será 6.

## Material necesario

Aproximadamente 30 cartulinas tamaño DIN A4, una para cada niño, y del mismo color. En cada una de las cartulinas se escribirá un número comprendido en un rango de 1 a 60.

## Metodología

El desarrollo de la actividad será de la siguiente manera:

Uno de los niños será el valor a buscar, el que "saltará" frente a sus compañeros. Estos se colocarán en orden de menor a mayor, respondiendo al número que tengan en la cartulina, tal y como correspondería a un array ordenado.

Los miembros del array tendrán la cartulina en sus manos de manera que el niño que busca no pueda ver el valor. Este niño que está buscando se colocará en frente del primero de sus compañeros. Le preguntará por su valor. El niño en el array desvelará el contenido de su cartulina, y aquí comienza el desarrollo de la técnica de búsqueda. Si es el número que está buscando, ya ha terminado, si no lo es, en este primer caso, porque el número revelado es menor al que se está buscando, el niño dará un salto de  $X$  posiciones.

El valor de  $X$  puede variar depende del número de niños. En el método de búsqueda original,  $X$  correspondería a la raíz cuadrada del tamaño, pero como no queremos complicar el desarrollo de la actividad lo explicaremos como la división entre 6 del tamaño del array.

Después de haber dado el salto, volverá a preguntar al niño que tiene en frente si es el valor que busca. Si es menor al esperado, volverá a dar un salto. Si es mayor, tendrá que preguntarle a todos

los niños a los que ha saltado en esta última iteración. Si por el contrario, es exactamente el valor que se está buscando, se daría por finalizada la búsqueda.

Para no complicar el desarrollo de la actividad, siempre se intentará que el valor a buscar esté entre los números que tienen los niños en las cartulinas.

### **Pasos a seguir**

1. Antes de desarrollar el juego, hay que preparar las cartulinas. En cada una de ellas se escribirá previamente un número.
2. A la llegada de los niños al aula, se le explicará detalladamente y lo más claro posible el desarrollo de la actividad.
3. Se le otorgará a cada uno, una cartulina. A continuación, se escogerá a uno de los niños para que sea el que busque el valor de su cartulina entre sus compañeros. Para esto, se pedirá un voluntario, si hay varios, se escogerá al azar.
4. Luego, se colocaran en orden de menor a mayor de acuerdo al número que tienen en sus láminas. Cuando el niño que busca les pregunta por su valor, deben revelar el número que tienen en la cartulina. El juego finaliza cuando se encuentra el valor.
5. Una vez terminado, dependiendo del tiempo que se haya empleado en la primera búsqueda, se repetirá la actividad.

### **Conclusiones**

Tras la finalización de esta actividad, no se pretende solamente que el público infantil al que va dirigida comprenda el término de búsqueda, sino que sean capaces de adaptar y comprender el pensamiento computacional, que se relaciona únicamente a la informática, a cualquier tipo de problema.

Además, haciendo llegar el pensamiento computacional a los niños, se les acerca este término desde una edad más temprana.

### **Agradecimientos**

Agradecemos al aula cultural de pensamiento computacional por esta iniciativa, que pretende hacer llegar la informática a los más pequeños.

### **Bibliografía**

1. Wikipedia - Jump Search
2. GeeksforGeeks - Traza Jump Search
3. Wikipedia - Pensamiento Computacional
4. ULL - Aula Cultural Pensamiento Computacional