

EJERCICIOS DE BÚSQUEDAS LINEAL Y BINARIA.

Curso 2019-2020

- 1.- Suponga que un vector ordenado de mayor a menor puede tener elementos repetidos. Implemente un algoritmo que usando la búsqueda binaria muestre todas las posiciones en las que se encuentra un determinado valor, si está.
- 2.- Se sabe que un vector de números de tipo `float` está ordenado de menor a mayor hasta la mitad y de mayor a menor desde la mitad hasta el final. Implemente un algoritmo que usando la búsqueda binaria encuentre la posición del valor más próximo a un entero dado como parámetro.
- 3.- Se sospecha que los elementos más buscados de una secuencia están al principio o al final de la secuencia. Implemente una búsqueda lineal desde ambos extremos.
- 4.- Considerar la búsqueda traspuesta en el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Muestra cómo queda el vector tras realizar la búsqueda traspuesta de cada uno de los valores siguientes (en este orden): 12, 20, 15, 20, 11, 20, 15, 20, 15, 25, 15, 10 y 15.
- 5.- Considerar la búsqueda con movimiento al frente en el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Muestra cómo queda el vector tras realizar la búsqueda de cada uno de los valores siguientes: 12, 20, 15, 20, 11, 20, 15, 20, 15, 25, 15, 10 y 15.
- 6.- Considerar el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Calcular cuántas comparaciones se hacen para buscar, con la búsqueda secuencial, 20 veces el 15 y después 20 veces el 25. ¿Y con la búsqueda traspuesta? ¿Y si se busca 20 veces el 15 y después el 25 (es decir, 15, 25, 15, 25, 15, ..., y así 20 veces)?
- 7.- Considerar la búsqueda binaria en este vector de 16 elementos
[12, 21, 24, 34, 37, 42, 45, 53, 67, 68, 70, 74, 77, 82, 88, 96]
 1. ¿Cuántas comparaciones hace la búsqueda lineal hasta encontrar 68?
 2. ¿Qué comparaciones hace la búsqueda binaria para buscar 68.
 3. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 68?
 4. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 53?
 5. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 21?
 6. ¿Cuántas comparaciones hace la búsqueda lineal hasta encontrar 21?
 7. ¿Y hasta encontrar 24, 34, 67, 30?
 8. ¿Cuál es el número máximo y mínimo de comparaciones hasta encontrar un valor (o no) en este vector?
 9. ¿Para cuántos valores la búsqueda binaria necesita dos comparaciones? ¿y cuatro?
 10. ¿Cuál es el número máximo de comparaciones hasta encontrar un valor?
- 8.- ¿Cuál es el número máximo y mínimo de comparaciones que realiza la búsqueda binaria hasta encontrar un valor (o no) si el vector, en vez de tener 8 elementos, tuviera 16? ¿y si tuviera 32? ¿y si tuviera 10, 20, 300, 100 o 1.000.000 elementos?

SOLUCIONES

1.- Suponga que un vector ordenado de mayor a menor puede tener elementos repetidos. Implemente un algoritmo que usando la búsqueda binaria muestre todas las posiciones en las que se encuentra un determinado valor, si está.

Solución:

```
int bus_binaria_rep(int x, int v[], int ini, int fin, int &primero, int &ultimo){
    pos = ini;
    while (ini < fin){
        pos = (ini + fin) / 2;
        if (v[pos] < x)
            ini = pos + 1;
        else
            fin = pos;
    }
    if (v[ini] == x){
        primero = ini
        while (v[primero] == x)
            primero--;
        primero++;
        ultimo = ini;
        while (v[ultimo] == x)
            ultimo++;
        ultimo--;
        return 0;
    }
    else return -1;
}
```

2.- Se sabe que un vector de números de tipo `float` está ordenado de menor a mayor hasta la mitad y de mayor a menor desde la mitad hasta el final. Implemente un algoritmo que usando la búsqueda binaria encuentre la posición del valor más próximo a un entero dado como parámetro.

Solución:

```
int bus_binaria2(int x, int v[], int ini, int fin){
    med = (ini + fin) / 2;
    pos = ini; fin2 = med ;
    while (ini < fin2){
        pos = (ini + fin2) / 2;
        if (v[pos] < x)
            ini = pos + 1;
        else
            fin = pos;
    }
    if (v[ini] == x){
        return ini;
    }
    else{
        ini = med + 1;
        med = (ini + fin) / 2;
        pos = ini;
        while (ini < fin){
            pos = (ini + fin) / 2;
            if (v[pos] < x)
                ini = pos + 1;
        }
    }
}
```

```

        else
            fin = pos;
    }
    if (v[ini] == x){
        return ini;
    }
    else return -1;
}

```

3.- Se sospecha que los elementos más buscados de una secuencia están al principio o al final de la secuencia. Implemente una búsqueda lineal desde ambos extremos.

Solución:

```

int bus_sec_doble(int x, int v[], int ini, int fin){
    int med = (ini + fin) / 2
    for (int pos = ini; pos <= med; pos++){
        if (v[pos] == x)
            return pos;
        if (v[fin-pos] == x)
            return fin-pos
    }
    return -1;
}

```

4.- Considerar la búsqueda traspuesta en el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Muestra cómo queda el vector tras realizar la búsqueda traspuesta de cada uno de los valores siguientes (en este orden): 12, 20, 15, 20, 11, 20, 15, 20, 15, 25, 15, 10 y 15.

Solución:

| Buscar | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Posición |
|--------|----|----|----|----|----|----|----|----|----------|
| 12 | 13 | 41 | 12 | 10 | 20 | 25 | 15 | 11 | 2 |
| 20 | 13 | 12 | 41 | 10 | 20 | 25 | 15 | 11 | 4 |
| 15 | 13 | 12 | 41 | 20 | 10 | 25 | 15 | 11 | 6 |
| 20 | 13 | 12 | 41 | 20 | 10 | 15 | 25 | 11 | 3 |
| 11 | 13 | 12 | 20 | 41 | 10 | 15 | 25 | 11 | 7 |
| 20 | 13 | 12 | 20 | 41 | 10 | 15 | 11 | 25 | 2 |
| 15 | 13 | 20 | 12 | 41 | 10 | 15 | 11 | 25 | 5 |
| 20 | 13 | 20 | 12 | 41 | 15 | 10 | 11 | 25 | 1 |
| 15 | 20 | 13 | 12 | 41 | 15 | 10 | 11 | 25 | 4 |
| 25 | 20 | 13 | 12 | 15 | 41 | 10 | 11 | 25 | 7 |
| 15 | 20 | 13 | 12 | 15 | 41 | 10 | 25 | 11 | 3 |
| 10 | 20 | 13 | 15 | 12 | 41 | 10 | 25 | 11 | 5 |
| 15 | 20 | 13 | 15 | 12 | 20 | 41 | 25 | 11 | 2 |
| | 20 | 15 | 13 | 12 | 20 | 41 | 25 | 11 | |

5.- Considerar la búsqueda con movimiento al frente en el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Muestra cómo queda el vector tras realizar la búsqueda de cada uno de los valores siguientes: 12, 20, 15, 20, 11, 20, 15, 20, 15, 25, 15, 10 y 15.

| Buscar | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Posición |
|--------|----|----|----|----|----|----|----|----|----------|
| 12 | 13 | 41 | 12 | 10 | 20 | 25 | 15 | 11 | 2 |
| 20 | 12 | 13 | 41 | 10 | 20 | 25 | 15 | 11 | 4 |
| 15 | 20 | 12 | 13 | 41 | 10 | 25 | 15 | 11 | 6 |
| 20 | 15 | 20 | 12 | 13 | 41 | 10 | 25 | 11 | 1 |
| 11 | 20 | 15 | 12 | 13 | 41 | 10 | 25 | 11 | 7 |
| 20 | 11 | 20 | 15 | 12 | 13 | 41 | 10 | 25 | 1 |
| 15 | 20 | 11 | 15 | 12 | 13 | 41 | 10 | 25 | 2 |
| 20 | 15 | 20 | 11 | 12 | 13 | 41 | 10 | 25 | 1 |
| 15 | 20 | 15 | 11 | 12 | 13 | 41 | 10 | 25 | 2 |
| 25 | 15 | 20 | 11 | 12 | 13 | 41 | 10 | 25 | 7 |
| 15 | 25 | 15 | 20 | 11 | 12 | 13 | 41 | 10 | 2 |
| 10 | 15 | 25 | 20 | 11 | 12 | 13 | 41 | 10 | 7 |
| 15 | 10 | 15 | 25 | 20 | 11 | 12 | 13 | 41 | 2 |
| | 15 | 10 | 25 | 20 | 11 | 12 | 13 | 41 | |

6.- Considerar el vector de 8 elementos [13, 41, 12, 10, 20, 25, 15, 11]. Calcular cuántas comparaciones se hacen para buscar, con la búsqueda secuencial, 20 veces el 15 y después 20 veces el 25. ¿Y con la búsqueda traspuesta? ¿Y si se busca 20 veces el 15 y después el 25 (es decir, 15, 25, 15, 25, 15, ..., y así 20 veces)?

Solución:

El 15 está en la posición 5 por lo que la búsqueda secuencial hace 6 comparaciones hasta encontrar el 15 cada vez. Por tanto son $6 \times 20 = 120$. Para buscar el 25 son $5 \times 20 = 100$. En total 220 comparaciones.

El número de comparaciones con la búsqueda traspuesta al buscar 20 veces el 15 depende de la posición del 15. El 15 estará una vez en la posición 6, otra en la 5, otra en la 4, otra .. así hasta 1 en la posición 1 y después el resto de las 15 veces en la posición 0. Por tanto el número de comparaciones es: $6 + 5 + 4 + 3 + 2 + 15 = 35$.

Después, el 25 empieza en la misma posición y por tanto son otras 35 comparaciones. 70 en total.

Si se busca 20 veces el 15 y después el 25, con la búsqueda secuencial, el número de comparaciones no varía; 220. Pero con la búsqueda traspuesta, el número buscado siempre está en la posición 6. Por tanto son $120 + 120 = 240$ comparaciones.

**7.- Considerar la búsqueda binaria en este vector de 16 elementos
[12, 21, 24, 34, 37, 42, 45, 53, 67, 68, 70, 74, 77, 82, 88, 96]**

1. ¿Cuántas comparaciones hace la búsqueda lineal hasta encontrar 68?
2. ¿Qué comparaciones hace la búsqueda binaria para buscar 68.
3. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 68?
4. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 53?
5. ¿Cuántas comparaciones hace la búsqueda binaria hasta encontrar 21?

6. ¿Cuántas comparaciones hace la búsqueda lineal hasta encontrar 21?
7. ¿Y hasta encontrar 24, 34, 67, 30?
8. ¿Cuál es el número máximo y mínimo de comparaciones hasta encontrar un valor (o no) en este vector?
9. ¿Para cuántos valores la búsqueda binaria necesita dos comparaciones? ¿y cuatro?
10. ¿Cuál es el número máximo de comparaciones hasta encontrar un valor?

Solución:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 12 | 21 | 24 | 34 | 37 | 42 | 45 | 53 | 67 | 68 | 70 | 74 | 77 | 82 | 88 | 96 |

1. El número de comparaciones hace la búsqueda lineal hasta encontrar 68 son 10 porque está en la posición 9.
2. Las comparaciones que hace la búsqueda binaria para buscar 68 son:
 busca en [0-15] compara con 53 en la posición $(0+15)/2=7$, $53 < 68$
 busca en [8-15] compara con 74 en la 11, $74 > 68$
 busca en [8-11] compara con 68 en la 9, $68=68$ y
 y finaliza devolviendo la posición 9
3. El número de comparaciones que hace la búsqueda binaria hasta encontrar 68 son 3
4. Las comparaciones que hace la búsqueda binaria hasta encontrar 53 es 1 porque $v[(0+15)/2] = 53$
5. Las comparaciones que hace hasta encontrar 21 son 3
6. La comparaciones que hace la búsqueda lineal hasta encontrar 21 son 2 porque está en la posición 1, la búsqueda binaria hace 4 comparaciones: con 53, 34, 21 y 12
7. La comparaciones que hace la búsqueda lineal hasta encontrar 24, 34, 67 y 30 son 3, 4, 9 y 16, respectivamente. Pero la búsqueda binaria hace 4, 2, 4 y 4 comparaciones, respectivamente.
8. El número máximo y mínimo de comparaciones que hace la búsqueda lineal hasta encontrar un valor (o no) en este vector es 16 y 1, respectivamente. La búsqueda binaria siempre hace 4, como máximo y 1 como mínimo.
9. La búsqueda binaria usa dos comparaciones para los valores de las posiciones $\lceil 15/4 \rceil = 3$, y $\lceil (3 \cdot 15)/4 \rceil = 11$, es decir para los valores 53 y 74. La búsqueda binaria usa 3 comparaciones para los valores de las posiciones $\lceil 15/8 \rceil = 1$, $\lceil (3 \cdot 15)/8 \rceil = 5$, $\lceil (5 \cdot 15)/8 \rceil = 9$ y $\lceil (7 \cdot 15)/8 \rceil = 13$, es decir para los valores 21, 42, 68 y 82. Y finalmente, usa 4 comparaciones para los valores de las posiciones pares: 0, 2, 4, 6, 8, 10, 12 y 14.
10. El número máximo de comparaciones que utiliza la búsqueda binaria hasta encontrar un valor en este vector es 4, que es el número de comparaciones para el valor de la posición 15, el 96. En la siguiente tabla se muestra el número de comparaciones para encontrar el valor en cada posición:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 4 | 3 | 4 | 2 | 4 | 3 | 4 | 1 | 4 | 3 | 4 | 2 | 4 | 3 | 4 | 5 |

8.- ¿Cuál es el número máximo y mínimo de comparaciones que realiza la búsqueda binaria hasta encontrar un valor (o no) si el vector, en vez de tener 8 elementos, tuviera 16? ¿y si tuviera 32? ¿y si tuviera 10, 20, 300, 100 o 1.000.000 elementos?

El número mínimo de comparaciones es 1 en todos los casos. El número máximo si el vector tiene 8 elementos es 4, si tiene 16 elementos es 5, si tiene 32 es 6. Nótese que $\log_2(8) = 3$, $\log_2(16) = 4$ y $\log_2(32) = 5$; o lo que es lo mismo $2^3 = 8$, $2^4 = 16$ y $2^5 = 32$. Para cualquier tamaño n que esté entre 8 y 16 (8), el número de comparaciones máximo será 4, y si el tamaño está entre 16 y 31 es 5. Por tanto, para 10 es 4, para 20 es 5, para 300 es 9, para 100 es 7 y para 1.000.000 es 20. En general es el menor entero mayor o igual que el logaritmo en base 2 del tamaño.