Algoritmos y estructuras de datos

Búsqueda lineal

Francisco Javier Zaragoza Martínez

Universidad Autónoma Metropolitana Unidad Azcapotzalco
Departamento de Sistemas

19 de julio de 2024

Citas

Eurípides

No dejes ninguna piedra sin remover.

Dichos populares

El que busca, encuentra. Para encontrar, primero debes buscar. Buscar aguja en pajar es naufragar. Cada cosa en su lugar ahorra tiempo al buscar.

Arthur C. Clarke

Todos los exploradores están buscando algo que perdieron. Es raro que lo encuentren, y más raro aún que el encontrarlo les traiga mayor felicidad que la búsqueda.

Problema abstracto

Dado un arreglo A con n elementos y un dato x, se desea saber si x está en el arreglo y, en ese caso, dónde está.

Problema concreto

Dado un arreglo `int a[MAX]` con `int n` elementos en las posiciones `a[0], ..., a[n-1]` y un dato `int x`, se desea saber si `x` está en el arreglo y, en ese caso, un índice `int i` tal que `a[i]==x`.

Búsqueda lineal

Implementación con arreglos

La búsqueda lineal consiste simplemente de preguntar en cada posición.

```
int  lineal(int n, int a[], int x) {
  for (int i = 0; i < n; i++)
    if (a[i] == x)
      return i;
  return -1;
}
```

En el peor de los casos se hacen n comparaciones (si x es el último o no está).

También lo podemos hacer con apuntadores:

```
int* lineal(int n, int *a, int x) {
  for (int *p = a; p < a+n; p++)
    if (*p == x)
      return p;
  return NULL;
}
```

En el peor de los casos se hacen n comparaciones (si x es el último o no está).

Ejercicios

1. Escribe una función recursiva `int lineal(int n, int a[], int x)` que lleve a cabo la búsqueda lineal en un arreglo desordenado.
2. Reescribe las dos versiones de `lineal` para comenzar a buscar desde el final del arreglo.
3. Reescribe las dos versiones de `lineal` para que regresen la cantidad de apariciones del elemento `x` en el arreglo.

Problemas de la búsqueda lineal

Estas funciones son muy lentas. En particular, si x no está en el arreglo entonces se requieren n comparaciones para darse cuenta.

¿Qué hacer?

- Si el arreglo no está ordenado no hay nada que hacer. ¿Por qué?
- ¿De qué sirve que el arreglo esté ordenado crecientemente?
- Veamos varias formas de aprovechar el orden.

Búsqueda lineal en un arreglo creciente

Implementación con arreglos

Cambiemos la condición de paro del ciclo:

```
int  lineal(int n, int a[], int x) {
  for (int i = 0; (i < n) && (a[i] <= x); i++)
    if (a[i] == x)
      return i;
  return -1;
}
```

Búsqueda lineal en un arreglo creciente

Implementación con apuntadores

También lo podemos hacer con apuntadores:

```
int* lineal(int n, int *a, int x) {
  for (int *p = a; (p < a+n) && (*p <= x); p++)
    if (*p == x)
      return p;
  return NULL;
}
```

Búsqueda lineal con saltos

Saltos de tamaño fijo

Una posibilidad para acelerar la búsqueda lineal es escoger un salto s y revisar las posiciones $A_0, A_s, A_{2s}, \ldots$ hasta que encontremos un elemento $A_{ks} \geq x$ o salgamos del arreglo. Una vez que esto pase, revisamos una por una las $\leq s$ posiciones que terminan en A_{ks} (o el fin del arreglo).

Búsqueda lineal con saltos

Saltos de tamaño fijo

Una posibilidad para acelerar la búsqueda lineal es escoger un salto s y revisar las posiciones $A_0, A_s, A_{2s}, \ldots$ hasta que encontremos un elemento $A_{ks} \geq x$ o salgamos del arreglo. Una vez que esto pase, revisamos una por una las $\leq s$ posiciones que terminan en A_{ks} (o el fin del arreglo).

Tiempo de ejecución

En la primera etapa se hacen $\leq 1 + \frac{n}{s}$ revisiones. En la segunda etapa se hacen $\leq s - 1$ revisiones adicionales. En total son $r(s) \leq s + \frac{n}{s}$ revisiones.

Búsqueda lineal con saltos

Saltos de tamaño fijo

Una posibilidad para acelerar la búsqueda lineal es escoger un salto s y revisar las posiciones $A_0, A_s, A_{2s}, \ldots$ hasta que encontremos un elemento $A_{ks} \geq x$ o salgamos del arreglo. Una vez que esto pase, revisamos una por una las $\leq s$ posiciones que terminan en A_{ks} (o el fin del arreglo).

Tiempo de ejecución

En la primera etapa se hacen $\leq 1 + \frac{n}{s}$ revisiones. En la segunda etapa se hacen $\leq s - 1$ revisiones adicionales. En total son $r(s) \leq s + \frac{n}{s}$ revisiones.

Mejor valor del salto

Tomando la derivada de $r(s)$ e igualando a cero obtenemos $0 = 1 - \frac{n}{s^2}$, es decir, $s = \sqrt{n}$. En este caso, $r(s) \leq 2\sqrt{n}$. ¡Esto es mucho mejor que lineal!

1. Escribe una función recursiva `int lineal(int n, int a[], int x)` que lleve a cabo la búsqueda lineal en un arreglo ordenado.
2. Escribe una función `int creciente(int n, int a[])` que decida si un arreglo cumple `a[0] <= a[1] <= ... <= a[n-1]`.
3. Escribe una función `int creciente(int n, int a[])` que decida si un arreglo cumple `a[0] < a[1] < ... < a[n-1]`.
4. Escribe una función `int pivote(int n, int a[], int p)` que decida si `a[i] <= a[p]` para toda `0 <= i < p` y `a[i] >= a[p]` para toda `p < i < n`.
5. Escribe una función `int salto(int n, int a[], int x, int s)` que implemente la búsqueda lineal con saltos de tamaño `s`.