



Sistemas Operativos

Sincronización de Threads

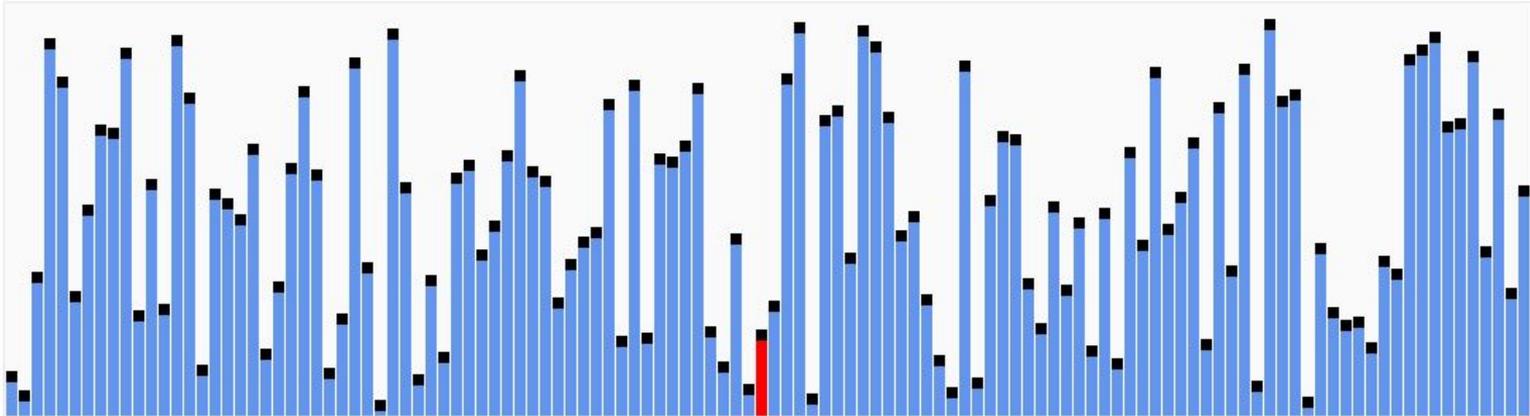
Diego Madariaga

1.

Quicksort paralelo

Quicksort

- ▷ Algoritmo de ordenamiento recursivo
- ▷ Se basa en la elección de un pivote en cada paso



Quicksort

```
void quicksort_seq(int a[], int i, int j) {  
    if (i < j) {  
        int h = particionar(a, i, j);  
        quicksort_seq(a, i, h-1);  
        quicksort_seq(a, h+1, j);  
    }  
}
```

Donde **particionar**:

- ▷ Selecciona el pivote
- ▷ Deja los elementos menores al pivote a su izquierda
- ▷ Deja los elementos mayores al pivote a su derecha
- ▷ Retorna la posición del pivote

Quicksort paralelo

- ▷ Utilizando N cores, programar:

```
void quicksort(int a[], int i, int j, int n);
```

2.

Colecta

Colecta

- ▷ Se necesita crear un sistema para juntar exactamente una cantidad X de dinero:
 - Definir el tipo de datos **Colecta**
 - Programar la función **Colecta *nuevaColecta(double meta)** que crea y retorna una colecta para juntar $\$meta$.
 - Programar la función **double aportar(Colecta *c, double monto)**, que es invocada desde múltiples threads para contribuir con $\$monto$. El valor de retorno de la función es el mínimo entre **monto** y lo que falta para llegar a la meta. La función debe retornar una vez que la meta se cumpla.



Sistemas Operativos

Sincronización de Threads

Diego Madariaga