


# Divide y Vencerás

## Conteo de inversiones

Norberto Garcia Gaspar  
Ángel Alberto Hamilton Lopez  
Óscar David Martín Cabrera



- Introducción
- Códigos
- Análisis de complejidad y tiempo de ejecución

# Introducción

- El conteo de inversiones pretende medir que tan desordenado esta un array.
- Se obtiene contando el número de movimientos necesarios para pasar del array original, a dicho array ordenado de menor a mayor o viceversa

# Solución mediante fuerza bruta

```
public static int countInversion(ArrayList<Integer> toAnalyze){  
    int inversionCount=0;  
  
    for(int i =0; i < toAnalyze.size(); i++)  
        for(int j =i; j < toAnalyze.size(); j++)  
            if(toAnalyze.get(i) >toAnalyze.get(j))  
                inversionCount++;  
  
    return inversionCount;  
}
```

# Solución Divide y Vencerás (1)

```
private ArrayList<Integer> countStep(ArrayList<Integer> lista, int izquierda, int derecha){
    ArrayList<Integer> result = new ArrayList<Integer>();

    //caso base en el que solo hay un elemento
    if(izquierda == derecha){
        result.add(lista.get(izquierda));
        return result;
    }

    //dividimos
    int mid = (izquierda + derecha)/2;
    ArrayList<Integer> izquierda =countStep(lista, izquierda, mid);
    ArrayList<Integer> derecha =countStep(lista, mid+1, derecha);
}
```

# Solución Divide y Vencerás (2)

```
//hacemos el merge
int i = izquierda, j = mid+1;
while((i <= mid) && (j <= derecha)){

    //como lo queremos ordenar de menor a mayor, si el menor esta a la derecha esta mal colocado
    if(derecha.get(j) < izquierda.get(i)){
        result.add(derecha.get(j));
        inversion++;
        j++;
    }
    else{
        result.add(izquierda.get(i));
        i++;
    }
}
```

## Solución Divide y Vencerás (3)

```
//añadimos los que sobran si sobran por algun lado
while(j < derecha){
    result.add(derecha.get(j));
    j++;
}
while(i < mid){
    result.add(izquierda.get(i));
    i++;
}
return result;
}
```

# Análisis de complejidad

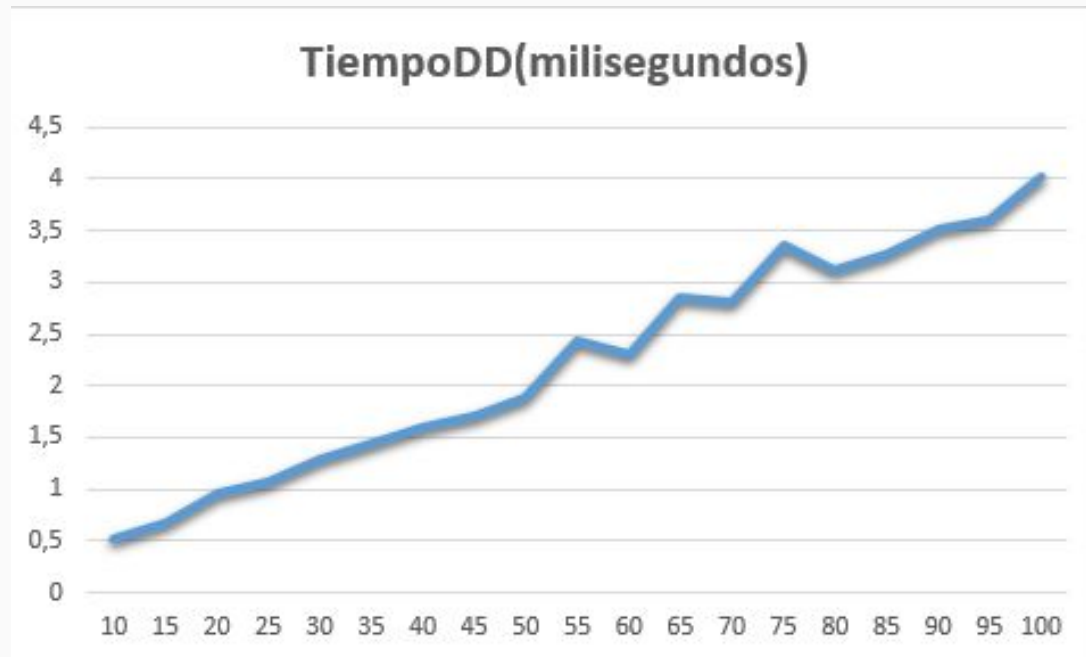
- El algoritmo de fuerza bruta es del orden  $\Theta(n^2)$
- El algoritmo de Divide y Vencerás es del orden  $\Theta(n \log_2(n))$



# Experimentación con fuerza bruta



# Experimentación con Divide y vencerás



# Comparación entre algoritmos

