Divide y vencerás:
Algoritmo de búsqueda de la mediana en un vector desordenado

Miguel Castro Caraballo
Carlos Troyano Carmona

Introducción

¿Qué es la mediana?

“La mediana representa el valor de la variable de posición central en un conjunto de datos ordenados.” - Wikipedia

Solo se puede hallar para variables cuantitativas.

¿Cómo se calcula?

Ordenar los datos. Si el número de datos es impar, coger el valor central, si no, coger los dos valores centrales y hacer el promedio.

Ejemplos de uso

Salario de un conjunto de personas

Descripción del problema

- Dado un array desordenado de n elementos, se quiere encontrar la mediana mediante la técnica de divide y vencerás.
- Este problema se puede abordar de dos formas
  - Ordenar todos los elementos de la secuencia primero y después determinar la mediana
  - No ordenar los elementos e intentar determinar la mediana mediante otro algoritmo

Algoritmos

Selection Algorithm - Quick Select

- Se elige un pivote y se colocan los números más pequeños a la izquierda y los más grandes a la derecha. Si el lugar donde está el pivote ahora no es n/2, repetimos el proceso por la rama que lo contiene.

Median of Medians

- Se dividen los datos en conjuntos de 5 elementos y se ordenan, después se halla la mediana de cada grupo y, finalmente, se halla la mediana de las medianas. Ese será el pivote.

Pseudocódigo

```
function select(V, left, right, n)
        if left = right
             return V[left]
        k := partition(V, left, right)
       lenght := k - izquierda + 1
        if lenght = k
            return V[k]
        else if n < k
            return select(V, n, left, k-1)
        else
            return select(V, n - leght, k + 1,
right)
```

Pseudocódigo

```
function partition(list, left, right)
    pivotValue := pivot(list, left, right)
    while(left < rigth)
      while(V[left] < pivotValue)    left++;
      while(V[right] > pivotValue)
right++;
     if(V[left] == V[right]) left++;
     else if swap V[left] and V[right]
return right;
```

Pseudocódigo

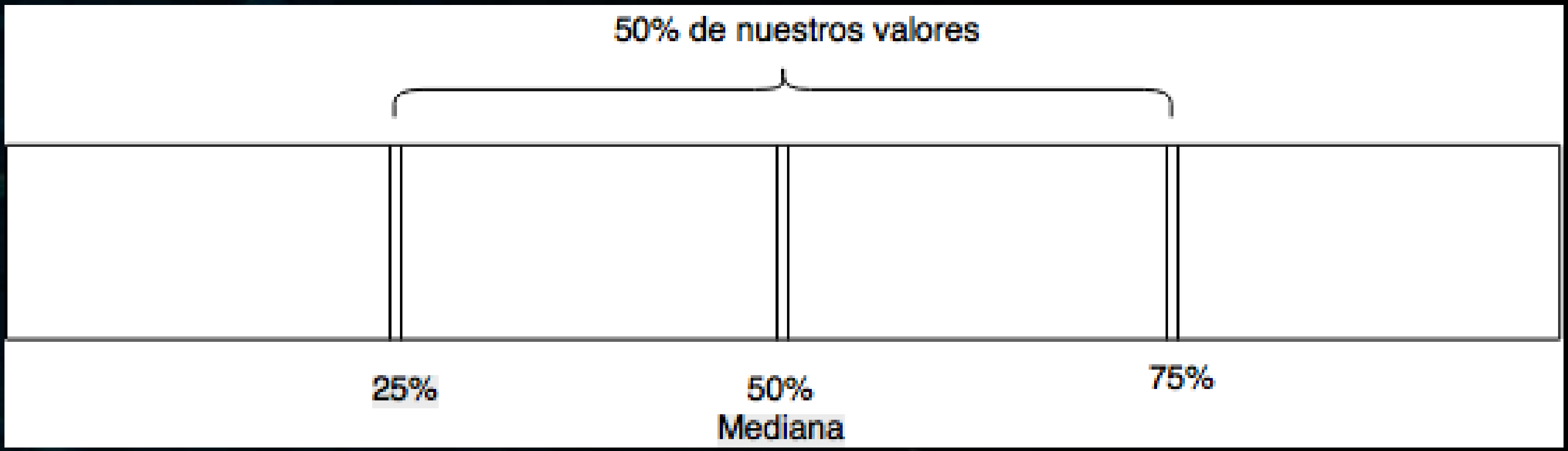
```
function pivot(V, left, right)
    // for 5 or less elements just get median
    if right - left + 1 < 9:
        sort(V)
        return V[V.length / 2]
    tmp as array;
    medians as array;
    medianIndex := 0;
    while (left <= right)
            tmp.size := MIN(5, right - left -1)
          for i from 0 to tmp.size AND left <=
right
                tmp[i] := V[left];
          left++;
      sort(tmp)
      medians[medianIndex] = tmp[tmp.size / 2];
      medianIndex++;
 return pivot(medians, 0, medians.size -1);
```

Análisis

Quick Selection tiene una complejidad de $O(n^2)$ (en el peor caso)

Analizaremos el Algoritmo Median of Medians

Elección de un pivote aleatoriamente



Análisis

- Tenemos el 50% de probabilidades de escoger un buen pivote y el 50% de no hacerlo por lo que representaremos dicha situación como ½. Por lo tanto. Si el tiempo esperado es Te(N) entonces:

  $Te(N) <= \frac{1}{2} Te(\frac{3}{4} N) + \frac{1}{2} Te(N) + O(n)$

  Restamos ½ Te(N) =

  $\frac{1}{2} Te(N) <= \frac{1}{2} Te(\frac{3}{4} N) + O(n)$

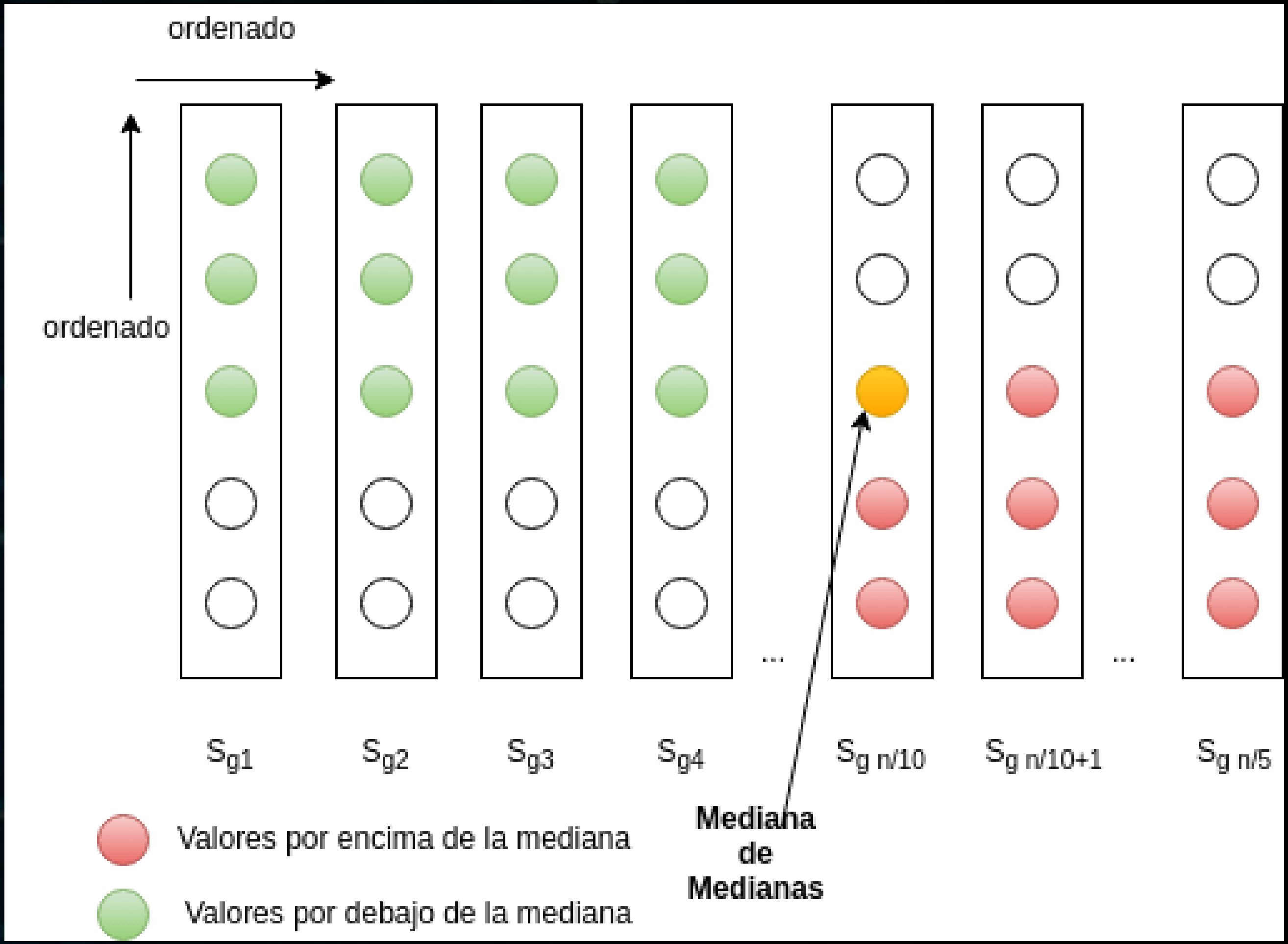
  Multiplicamos por 2 =

  $Te(N) <= Te(\frac{3}{4} N) + O(n)$

  Por el teorema maestro sabemos que el tiempo esperado es el caso 1 O(n).

Análisis

Subdivisión de conjuntos de 5 elementos.



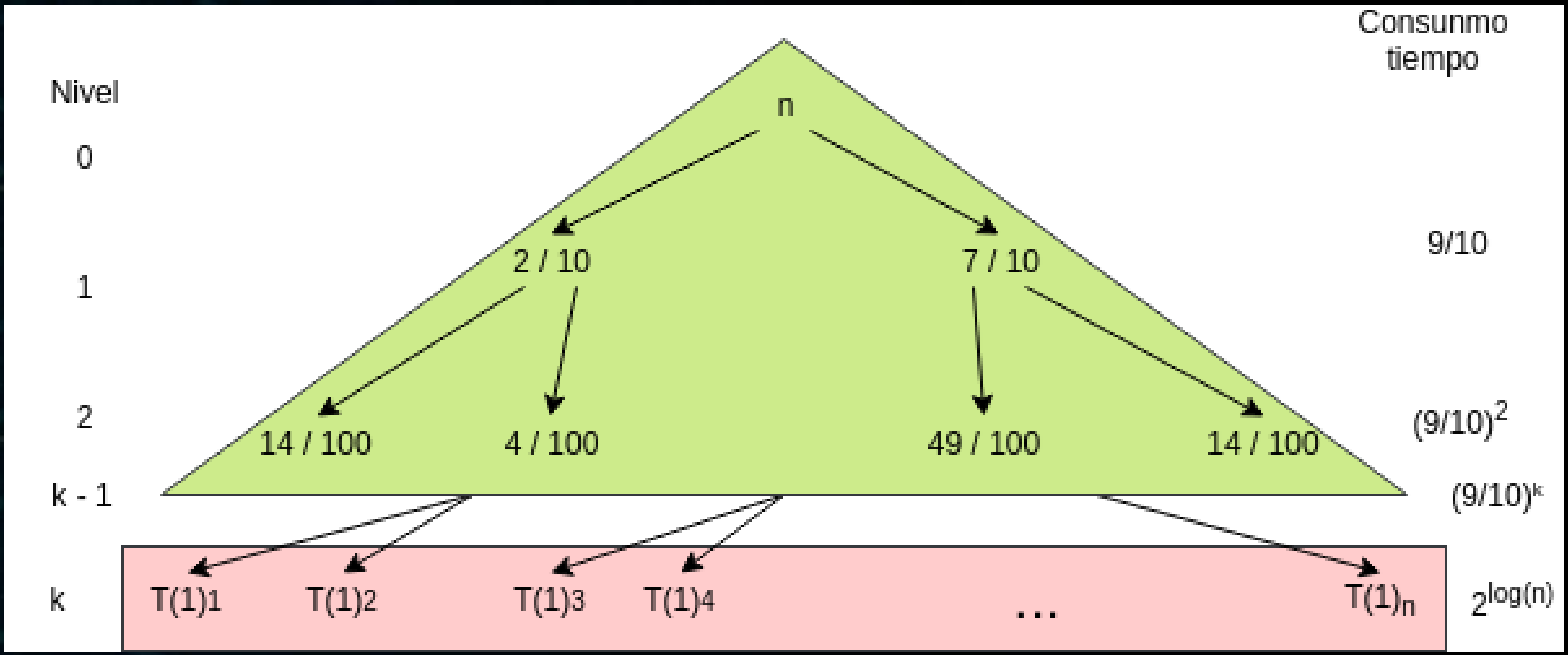
Análisis

$$T(n) = T(n7/10) + T(n/5) + O(n)$$

- **T(n7/10) : Tiempo en la recursión del método principal.**
- **T(n/5) : Tiempo para hallar la media de las medias.**
- **O(n) : Tiempo en subdividir el problema**
- **Todo peor caso.**

Análisis

Árbol de recursión

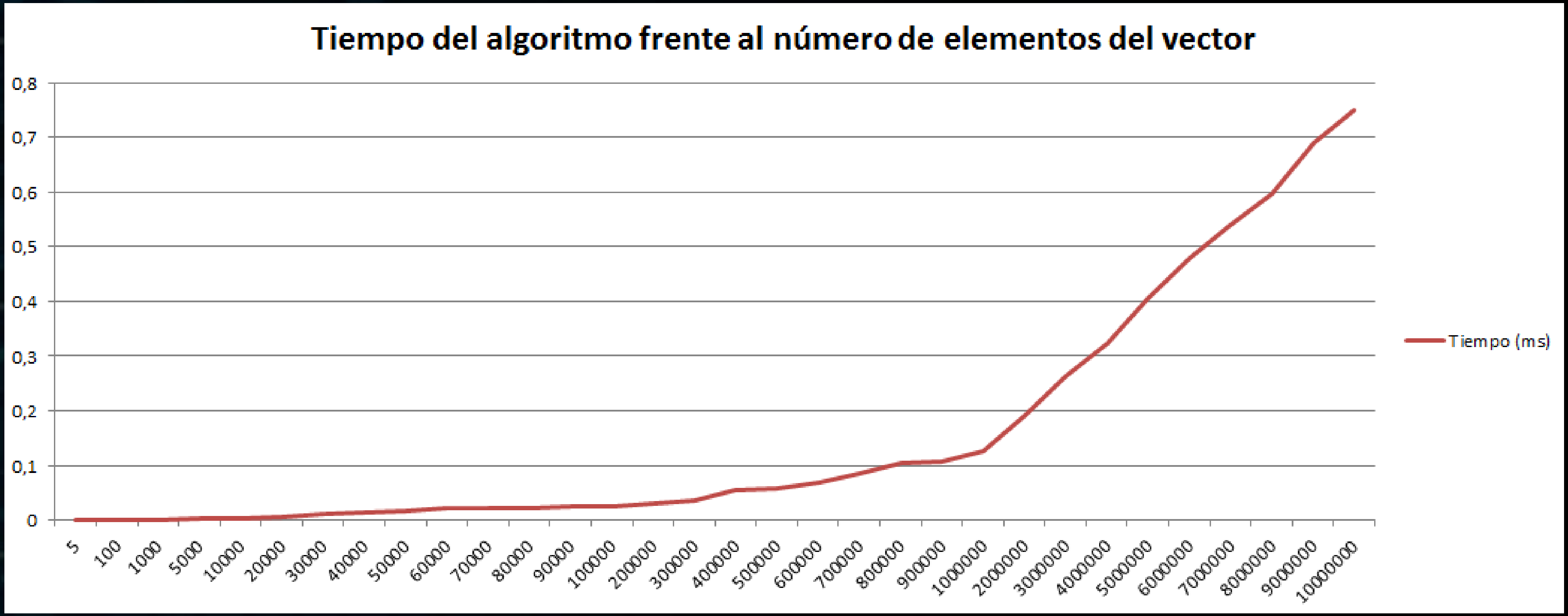


Análisis

- Como vemos el árbol tiene un tamaño de k niveles donde la altura k en el peor caso en n.
- Por lo que la suma de todos los casos base es 2log(n)C y el resto del árbol es SUM desde i=0 hasta n de $((9/10)^i)$
- Resolvemos la progresión aritmética con la fórmula. 1 / 1 - (9 / 10) = 10.
- Entonces $T(n) = 2^{\log(n)}C + 10C + O(n) \Rightarrow T(n) = O(n)$
- Por lo que llegamos a la conclusión de que el algoritmo es de tiempo polinomial O(n).

Evaluación experimental

- Realizando 30 pruebas (con distinto rango), se obtiene una gráfica de la siguiente forma:



Conclusiones

A la vista de los resultados obtenidos se pueden extraer las siguientes conclusiones:

- Las pruebas demuestran de forma empírica que el algoritmo “Median of Medians” resuelve el problema en tiempo lineal. En la misma gráfica se aprecia la tendencia de los valores a formar una línea recta.
- El algoritmo “Median of Medians” es un claro ejemplo de algoritmo que, dada una gran cantidad de datos, obtiene un resultado en un tiempo eficiente.

Bibliografía

- https://en.wikipedia.org/wiki/Quickselect (Wikipedia)
- https://en.wikipedia.org/wiki/Median_of_medians (Wikipedia)
- http://cs.indstate.edu/~spitla/presentation.pdf (Indiana State University)
- http://www.vitutor.com/estadistica/descriptiva/a_9.html (Vitutor)