Universidad Nacional de Ingeniería
Ciencias de la Computación
Quicksort

Yuri Nuñez Medrano *
ynunezm@gmail.com

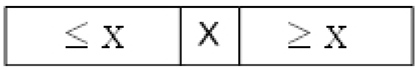

Figura 1: división

Resumen

Se analizará algortimos tipo Quicksort.

1. Analisis del Quicksort

Analizaremos el Quicksort y el Randomized Quicksort

1.1. Descripción del Quicksort

Evaluamos el algoritmo con el método divide and conquer.
Ordenar un Array
1. División: Realizar particiones (partition en el algoritmo 1). en 2 subarrays, alrededor de un valor (pivote) de tal manera que los elementos menores $\leq x \leq$, en la figura 1

2. Conquistar: recursivamente en dos subarrays.
3. Combinar: Trivial.

Algorithm 1: PARTITION(A,p,r)

```
x=A[r]
i=p-1
for j=p to r-1 do
    if A[j] ≤ x then
        i=i+1
        exchange A[i] with A[j]
exchange A[i+1] with A[r]
return i+1
```

Algorithm 2: QUICKSORT(A,p,r)

```
if p<r then
    q=PARTITION(A,p,r)
    QUICKSORT(A,p,q-1)
    QUICKSORT(A,q+1,r)
```

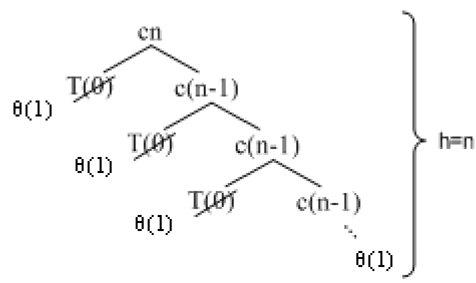

Figura 2: división

1.2. Analisis del caso peor

El tiempo de ejecución cuando el input esta ascendente o descenente, cuando una partición no tiene elementos.
$T(n) = T(0) + T(n-1) + \Theta(n)$.
$\Theta(1) + T(n-1) + \Theta(n)$.
$T(n-1) + \Theta(n)$.
$\Theta(n^2)$.
La complejida ser parecida al INSERTION_SORT.
Utilizando el arbol recursivo.
en la figura 2.

$\Theta(\sum_{k=1}^{n} ck)$.
$\Theta(n^2)$.

1.3. Analisis del caso mejor

Si tenemos suerte, en la partición de array de $n/2n/2$ entonces nos quedaria.
$T(n) = 2T(n/2) + \Theta(n)$

*Escuela de Ciencias de la Computación, 27-08-15

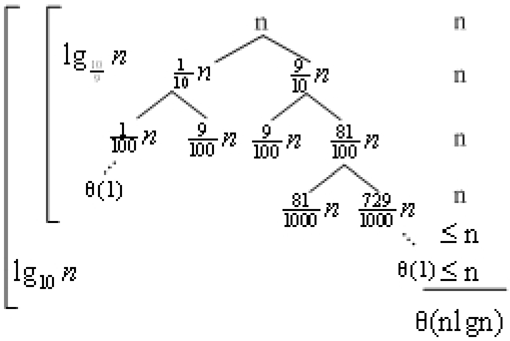

Figura 3: arbol de recursion

$= \Theta(nlgn)$
Si tuvieramos el caso de una pertición proporcianal de la siguiente forma.
$T(n) = \Theta(n) + T(\frac{a-1}{a}n) + T(\frac{1}{a}n), para\ a > 2.$
Ejm:
$T(n) = \Theta(n) + T(\frac{9}{10}n) + T(\frac{1}{10}n)$
seria como en la figura 3.

$cnlog_{10}n \leq T(n) \leq cnlog_{10/9}n$
Suponinedo que alternamos entre un caso luky(suertudo) y uno unluky(sin suerte). Tendriamos las siguientes ecuaciones:
$L(n) = 2U(n/2) + \Theta(n).$
$U(n) = L(n-1) + \Theta(n).$
Sustituimos.
$L(n) = 2[L(\frac{n}{2} - 1) + \Theta(\frac{n}{2})]$
$2L(\frac{n}{2} - 1) + \Theta(n)$
$\Theta(nlgn)$

1.4. Randomized Quicksort

El tiempo de ejecución es independiente del ordenamiento del input. Sin suposiciones del input. El caso peor solo es determinado por el generador de numeros aleatorios (el pivot es aleatorio).

Algorithm 3: RANDOMIZED_PARTITION(A,p,r)

```
i=RANDOM(p,r)
exchange A[r] with A[i]
return PARTITION(A,p,r)
```

T(n): variable aleatoria para un tiempo de ejecución se asume números aleatórios independientes.
for $k = 0, 1, ..., n-1$

Algorithm 4: RANDOMIZED_QUICKSORT(A,p,r)

```
if p<r then
    q=RANDOMIZED_PARTITION(A,p,r)
    RANDOMIZED_QUICKSORT(A,p,q-1)
    RANDOMIZED_QUICKSORT(A,q+1,r)
```

$$X_k = \left\{ \begin{array}{ll} 1 & si\ genera\ part.\ k : n-k-1 \\ 0 & para\ los\ demas \end{array} \right\} \quad (1)$$

Evaluamos la Espectativa (Esperanza).

$E[X_k] = 0.Pr\{X_k = 0\} + 1.Pr\{X_k = 1\}$
$= Pr\{X_k = 1\}$
$= 1/n$

$$T(n) = \left\{ \begin{array}{ll} T(0) + T(n-1) + \Theta(n) & si\ 0 : n-1 \\ T(1) + T(n-2) + \Theta(n) & si\ 1 : n-2 \\ ... & ... \\ T(n-1) + T(0) + \Theta(n) & si\ n-1 : 0 \end{array} \right\}$$

$E[T(n)] = E[\sum ...]$
$= \sum_{k=0}^{n-1} E[X_k(T(k) + T(n-k-1) + \Theta(n))]$
$= (\underbrace{\sum_{k=0}^{n-1} E[X_k]}_{\frac{1}{n}}) \ (\sum_{k=0}^{n-1} E[(T(k) + T(n-k-1) + \Theta(n))])$

$= \frac{1}{n}\sum_{k=0}^{n-1} E[(T(k) + T(n-k-1)] + \frac{1}{n}\sum_{k=0}^{n-1} E(\Theta(n))$

$= \frac{2}{n}\sum_{k=0}^{n-1} E[T(k)] + \Theta(n)$
Absorbe $k = 0, 1$ en terminos de $\Theta(n)$ por conveniencia.

Probamos $E[T(n)] \leq anlgn$ para ctes $a > 0$ escoger un "a" suficientemente grande
$anlgn \geq E[T(n)]$ Asumimos.
$\sum_{k=2}^{n-1} klgk \leq \frac{1}{2}n^2lgn - \frac{1}{8}n^2$
por substitución.
$E[T(n)] \leq \frac{2}{n}\sum_{k=2}^{n-1} aklgk + \Theta(n)$
$\leq \frac{2a}{n}(\frac{1}{2}n^2lgn - \frac{1}{8}n^2) + \Theta(n)$
$= \underbrace{anlgn}_{requerido} - \underbrace{(\frac{an}{4} - \Theta(n))}_{residuo\,positivo}$

Referencias

[H.Cormen et al., 2009] H.Cormen, T., Leiserson, C., and Riverson, R. L. (2009). *Algorithms*. The MIT Press.