**QuickSort**

É um algoritmo do tipo Divide & Conquista.

**Divide:** Reorganize os elementos e divida as matrizes em duas submatrizes e um elemento no meio procure que cada elemento na submatriz esquerda seja menor ou igual ao elemento médio e cada elemento na submatriz direita seja maior que o elemento do meio .

**Conquistar:** Recursivamente, classifique duas submatrizes.

**Combine:** Combine a matriz já classificada.

Algoritmo:

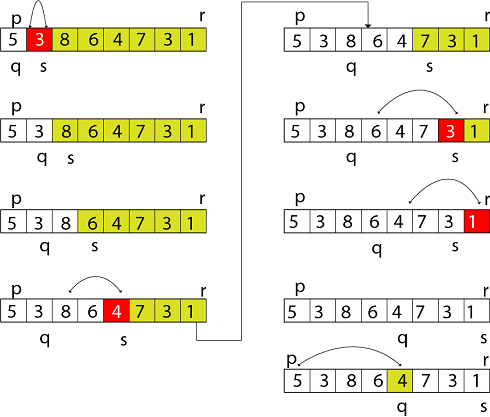
1. QUICKSORT (matriz A,  **int** m,  **int** n)
2. 1 **se** (n> m)
3. 2  então
4. 3  i ← um índice aleatório de [m, n]
5. 4  troca A [i] com A [m]
6. 5  o ← PARTIÇÃO (A, m, n)
7. 6  QUICKSORT (A, m, o -  1 )
8. 7  QUICKSORT (A, o +  1 , n)

Algoritmo de partição:

O algoritmo de partição reorganiza as submatrizes em um local.

1. PARTITION (matriz A,  **int** m,  **int** n)
2. 1  x ← A [m]
3. 2  o ← m
4. 3 **para** p ← m +  1  para n
5. 4 **faça if** (A [p] <x)
6. 5  depois o ← o +  1
7. 6  troca A [o] com A [p]
8. 7  troca A [m] com A [o]
9. 8 **retorno** o

**Figura: mostra o algoritmo de partição de rastreamento de execução**



Exemplo de classificação rápida:

1. 44 33 11 55 77 90 40 60 99 22 88

Seja **44** o elemento **Pivo** e a varredura feita da direita para a esquerda

Comparando **44** com os elementos do lado direito e se os elementos do lado direito forem **menores** que **44** , troque-os. Como **22** é menor que **44** , troque-os.

**22** 33 11 55 77 90 40 60 99 **44** 88

Agora, comparando **44** com o elemento do lado esquerdo e o elemento deve ser **maior** que 44, troque-os. Como **55** são maiores que **44** , troque-os.

22 33 11 **44** 77 90 40 60 99 **55** 88

Recursivamente, repetindo as etapas 1 e 2 até obtermos duas listas, uma à esquerda do elemento pivô **44** e uma à direita do elemento pivô.

22 33 11 **40** 77 90 **44** 60 99 55 88

**Trocar com 77:**

22 33 11 40 **44** 90 **77** 60 99 55 88

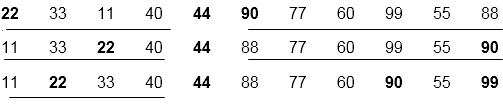
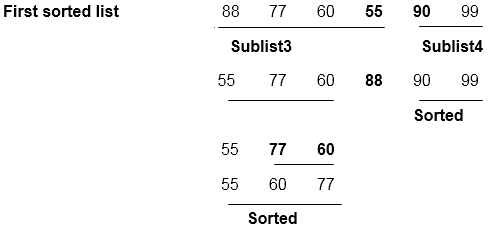
Agora, o elemento do lado direito e do lado esquerdo são maiores e menores do que **44,** respectivamente.

**Agora temos duas listas classificadas:**

Classificação rápida DAA

E essas sublistas são classificadas sob o mesmo processo feito acima.

Essas duas sublistas classificadas lado a lado.

Mesclando Sublistas:

Classificação rápida DAA

**LISTAS SORTED**

**Análise do pior caso:** é o caso quando os itens já estão classificados e tentamos classificá-los novamente. Isso leva muito tempo e espaço.

Equação:

1. T (n) = T ( 1 ) + T (n- 1 ) + n

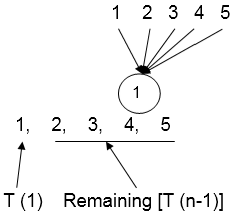
**T (1)** é o tempo gasto pelo elemento pivô.

**T (n-1)** é o tempo gasto pelo elemento restante, exceto pelo elemento pivô.

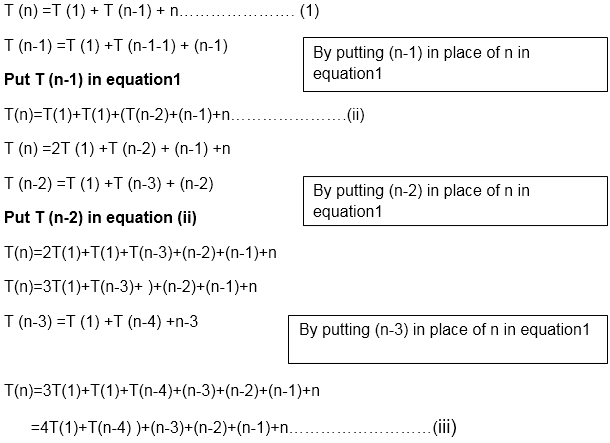
**N:** o número de comparações necessárias para identificar a posição exata de si mesmo (cada elemento)

Se compararmos o pivô do primeiro elemento com outro, haverá 5 comparações.

Isso significa que haverá n comparações se houver n itens.



Fórmula relacional para o pior caso:



**Nota: para fazer T (n-4) como T (1) colocaremos (n-1) no lugar de '4' e se  
colocarmos (n-1) no lugar de 4, então teremos que colocar (n- 2) no lugar de 3 e (n-3)  
No lugar de 2 e assim por diante.**

T (n) = (n-1) T (1) + T (n- (n-1)) + (n- (n-2)) + (n- (n-3)) + (n- ( n-4)) + n  
T (n) = (n-1) T (1) + T (1) + 2 + 3 + 4 + ............ n  
T (n) = (n-1) T (1) + T (1) + 2 + 3 + 4 + ........... + n + 1-1

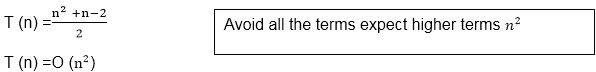
[Adicionando 1 e subtraindo 1 para fazer a série AP]

T (n) = (n-1) T (1) + T (1) + 1 + 2 + 3 + 4 + ........ + n-1  
T (n) = (n-1) T (1) + T (1) + Classificação rápida DAA-1

**Condição de parada: T (1) = 0**

Porque, finalmente, resta apenas um elemento e nenhuma comparação é necessária.

T (n) = (n-1) (0) +0+ Classificação rápida DAA-1



**A complexidade do pior caso de classificação rápida é T (n) = O (n 2 )**

Classificação rápida aleatória [caso médio]:

Geralmente, assumimos o primeiro elemento da lista como o elemento pivô. Em um caso médio, o número de chances de obter um elemento pivô é igual ao número de itens.

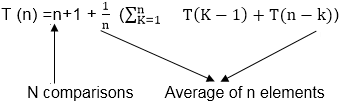
1. Seja o tempo total gasto = T (n)
2. Por exemplo: Em uma determinada lista
3. p  1 , p  2 , p  3 , p  4 ............ pn
4. Se p  1  é a lista dinâmica, então temos  2  listas.
5. Ou seja, T ( 0 ) e T (n- 1 )
6. Se p2 for a lista dinâmica, então temos  2  listas.
7. Ou seja, T ( 1 ) e T (n- 2 )
8. p  1 , p  2 , p  3 , p  4 ............ pn
9. Se p3 for a lista dinâmica, então temos  2  listas.
10. Ou seja, T ( 2 ) e T (n- 3 )
11. p  1 , p  2 , p  3 , p  4 ............ p n

Portanto, em geral, se considerarmos o elemento **K** - **ésimo** como o elemento pivô.

**Então,**

Classificação rápida DAA

O elemento Pivot fará uma comparação n e estamos fazendo um caso médio, então,



**Portanto, a fórmula relacional para classificação rápida aleatória é:**

Classificação rápida DAA

= n + 1 + Classificação rápida DAA(T (0) + T (1) + T (2) + ... T (n-1) + T (n-2) + T (n-3) + ... T ( 0))

= n + 1 + Classificação rápida DAAx2 (T (0) + T (1) + T (2) + ... T (n-2) + T (n-1))

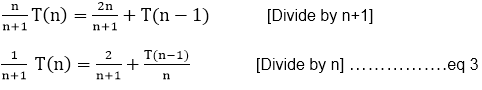
1. n T (n) = n (n + 1 ) + 2   (T ( 0 ) + T ( 1 ) + T ( 2 ) + ... T (n- 1 ) ........ eq  1

Coloque n = n-1 na eq 1

1. (n - 1 ) T (n - 1 ) = (n - 1 ) n + 2  (T ( 0 ) + T ( 1 ) + T ( 2 ) + ... T (n - 2 ) ...... eq2

De eq1 e eq 2

n T (n) - (n-1) T (n-1) = n (n + 1) -n (n-1) +2 (T (0) + T (1) + T (2) +? T (n-2) + T (n-1)) - 2 (T (0) + T (1) + T (2) + ... T (n-2))  
n T (n) - (n -1) T (n-1) = n [n + 1-n + 1] + 2T (n-1)  
n T (n) = [2+ (n-1)] T (n-1) + 2n  
n T (n) = n + 1 T (n-1) + 2n



Coloque n = n-1 na eq 3

Classificação rápida DAA

Coloque 4 eq em 3 eq

Classificação rápida DAA

Coloque n = n-2 na eq 3

Classificação rápida DAA

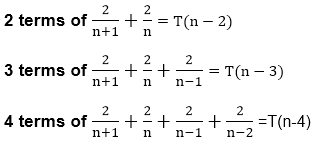
Coloque 6 eq em 5 eq

Classificação rápida DAA

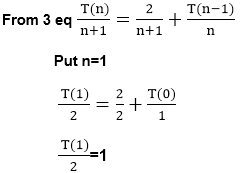
Coloque n = n-3 na eq 3

Classificação rápida DAA

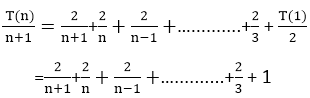
Coloque 8 eq em 7 eq

Classificação rápida DAA  


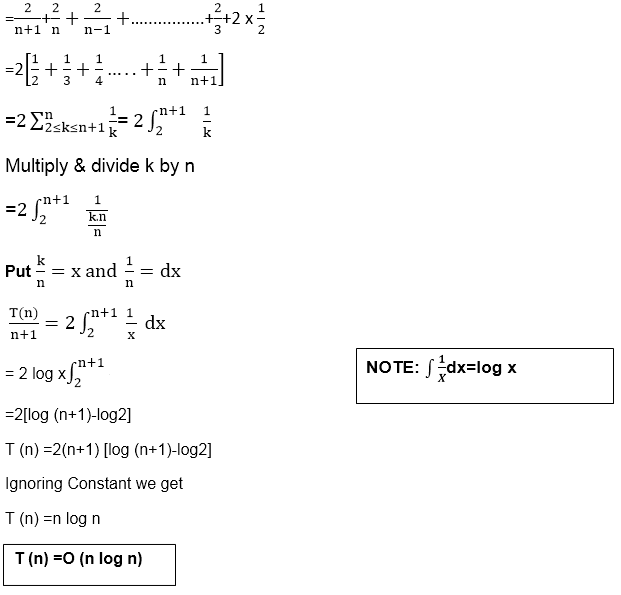
De 3eq, 5eq, 7eq, 9 eq, obtemos

Classificação rápida DAA  


De 10 eq



Multiplique e divida o último termo por 2



**É a complexidade média do caso de classificação rápida para classificar n elementos.**

**3. Classificação rápida [Melhor caso]:** Em qualquer classificação, o melhor caso é o único caso em que não fazemos nenhuma comparação entre elementos que só é feita quando temos apenas um elemento para classificar.

