Analysis of Selection Sort

Const		freq.
	Algorithm Selection_Sort (A, n)	
	{	
C_1	for i = 1 to n do	$n+1$
	{	
C_2	j = i; // will eventually point to min element	n
C_3	for k = i+1 to n do	$\sum_{i=1}^{n} (n-i+1)$
	{	
C_4	if (A[k] < A[j]) then	$\sum_{i=1}^{n} (n-i)$
	{	
C_5	j = k ;	$\sum_{i=1}^{n} (n-i)\, t_i$
	}	
	}	
C_6	t = A[i] ;	n
C_7	A[i] = A[j] ;	n
C_8	A[j] = t ;	n
	}	
	}	

Constant column – time for once time execution of associated line

Frequency – No. of times associated line will be executed.

t_i – Represents boolean function which is true or false for that i^{th} iteration.

Best Case Analysis

- List is already sorted.
- So, boolean function $t_i = 0$ always, if condition fails always.

$\therefore$ assignment $j = k$ will never be executed.

Evaluating Summations,

$$T(n) = \sum_{i=1}^{n}(n-i+1) = \sum_{i=1}^{n} n - \sum_{i=1}^{n} i + \sum_{i=1}^{n} 1$$

$$= n\sum_{i=1}^{n} 1 - \frac{n(n+1)}{2} + n$$

$$= n^2 - \frac{n(n+1)}{2} + n$$

$$= n^2 - \left(\frac{n^2+n}{2}\right) + n$$

$$= \frac{n^2+n}{2} = \frac{n(n+1)}{2} \quad \ldots (\mathrm{I})$$

$$\sum_{i=1}^{n}(n-i) = \sum_{i=1}^{n} n - \sum_{i=1}^{n} i$$

$$= n^2 - \frac{n(n+1)}{2}$$

$$= n^2 - \frac{(n^2+n)}{2}$$

$$= \frac{n^2-n}{2} = \frac{n(n-1)}{2} \quad \ldots (\mathrm{II})$$

Using Summations,

$$T(n) = c_1(n+1) + c_2 n + c_3 \frac{n(n+1)}{2} + c_4 \frac{n(n-1)}{2} +$$

$$c_6 n + c_7 n + c_8 n$$

$$= c_1 n + c_1 + c_2 n + \frac{c_3}{2} n^2 + \frac{c_3}{2} n + \frac{c_4}{2} n^2 - \frac{c_4}{2} n +$$

$$c_6 n + c_7 n + c_8 n$$

$$= \left[\frac{c_3}{2} + \frac{c_4}{2}\right] n^2 + \left[c_1 + c_2 + \frac{c_3}{2} - \frac{c_4}{2} + c_6 + c_7 + c_8\right] n$$

$$+ \{c_1$$

By considering constant as unit value,

$$T(n) = n^2 + 5n + 1$$

This is a quadratic function.

$$\therefore \boxed{T(n) = \Theta(n^2)}$$

Worst Case Analysis

- List is in reverse order.
- In this case, t_i will always be True,
  $\therefore$ assignment statement $j = k$ will always get executed.

Using expressions (I) & (II)

$$T(n) = c_1(n+1) + c_2 n + c_3 \frac{n(n+1)}{2} + c_4 \frac{n(n-1)}{2} + c_5 \frac{(n(n-1))}{2}$$

$$+ c_6 n + c_7 n + c_8 n$$

$$= c_1 n + c_1 + c_2 n + \frac{c_3}{2} n^2 + \frac{c_3}{2} n + \frac{c_4}{2} n^2 - \frac{c_4 n}{2} + \frac{c_5}{2} n^2 - \frac{c_5 n}{2}$$

$$+ c_6 n + c_7 n + c_8 n$$

$$= \left[\frac{c_3}{2} + \frac{c_4}{2} + \frac{c_5}{2}\right] n^2 + \left[c_1 + c_2 + \frac{c_3}{2} - \frac{c_4}{2} - \frac{c_5}{2} + c_6 + c_7 + c_8\right] n + c_1$$

By considering constant as unit value,

$$T(n) = 1.5n^2 + 4.5n + 1$$

This is a quadratic function

Hence $\boxed{T(n) = \Theta(n^2)}$

Average Case Analysis

- Half of the list is sorted and half is not.
- In this case, t_i will be true for half of times.

$\therefore$ assignment statement $j = k$ will be executed $\frac{n-i}{2}$ time for each i^{th} iteration.

$$\therefore \sum_{i=1}^{n} \frac{n-i}{2} = \frac{1}{2}\left[\sum_{i=1}^{n} n - \sum_{i=1}^{n} i\right]$$

$$= \frac{1}{2}\left[n^2 - \frac{n(n+1)}{2}\right]$$

$$= \frac{1}{2} n^2 - \frac{n^2}{4} + \frac{n}{4}$$

$$\sum_{i=1}^{n} \frac{n-i}{2} = \frac{n^2}{2} + \frac{n}{4}$$

$\therefore$ Similar to worst case, for average case, time taken will be quadratic function of input.

$$\therefore T(n) = \Theta(n^2)$$

Key = 3

| 5 | 3 | 4 | 1 | 2 |

| 5 | 3 | 4 | 1 | 2 |

Analysis of Insertion Sort

Const		freq
	Algorithm Insertion_Sort (A, n)	
	{	
c_1	for j = 2 to n do // Array indexing is from 1 to n	n
	{	
c_2	key = A[j];	$n-1$
c_3	i = j - 1; // index for sorted section	$n-1$
c_4	while (i > 0 AND A[i] > key)	$\sum_{j=2}^{n} t_j$
	{	
c_5	A[i+1] = A[i]; // Right shift	$\sum_{j=2}^{n}(t_j - 1)$
c_6	i = i - 1;	$\sum_{j=2}^{n}(t_j - 1)$
	}	
c_7	A[i+1] = key //inserting element	$n-1$
	}	
	}	

Constant Column - represents the time required to execute the associated line for once

Frequency - represents no. of time the associated line will be executed.

t_j – for j^{th} iteration, No. of times line will be executed

Best Case Analysis

Best case : when the array / list is already sorted.

- In this, while condition fails always. So while line will be executed $(n-1)$ time.
- And statements inside will loop will never be executed.

Total time required will be,

$$T(n) = c_1 * n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1)$$

$$= (c_1 + c_2 + c_3 + c_4 + c_7)n - (c_2 + c_3 + c_4 + c_7)$$

If we assume all constants have unit value, then,

$$\boxed{T(n) = 5n - 4}$$

This is a linear function of n,

$\therefore$ $T(n) = \Theta(n)$

$$\boxed{n \le 5n-4 \le 5n}$$

$n \geq 1$

Worst Case Analysis

Worst Case : when the given array/list is in descending/reverse order.

- In this, while condition is true always.
- For each iteration j, there will be j comparisons.

∴ we can say that,

$$\sum_{j=2}^{n} t_j = \sum_{j=2}^{n} j$$

$$= \frac{n(n+1)}{2} - 1$$

$$\sum_{j=2}^{n} (t_j - 1) = \sum_{j=2}^{n} (j-1) = \sum_{j=2}^{n} j - \sum_{j=2}^{n} 1$$

$$= \frac{n(n+1)}{2} - 1 - (n-1)$$

$$= \frac{n^2+n}{2} - n$$

$$= \frac{n^2+n-2n}{2}$$

$$= \frac{n^2-n}{2}$$

$$= \frac{n(n-1)}{2}$$

using summations,

$$T(n) = c_1 \cdot n + c_2(n-1) + c_3(n-1) + c_4\left(\frac{n(n+1)}{2} - 1\right) + c_5\left(\frac{n(n-1)}{2}\right) + c_6\left(\frac{n(n-1)}{2}\right) + c_7(n-1)$$

$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + \frac{c_4 n^2}{2} + \frac{c_4 n}{2} - c_4 + \frac{c_5 n^2}{2} - \frac{c_5 n}{2} + \frac{c_6 n^2}{2} - \frac{c_6 n}{2} + c_7 \cdot n - c_7$$

$$= \left[\frac{c_4}{2} + \frac{c_5}{2} + \frac{c_6}{2}\right] n^2 + \left[c_1 + c_2 + c_3 + \frac{c_4}{2} - \frac{c_5}{2} - \frac{c_6}{2} + c_7\right] n - \left[c_2 + c_3 + c_4 + c_7\right]$$

By considering all constant have unit value,

$$T(n) = 1.5\, n^2 + 3.5\, n - 4$$

This is a quadratic function,

Hence, we can say that

$$\boxed{T(n) = \theta(n^2)}$$

$$n^2 \leq 1.5n^2 + 3.5n - 4 \leq 5n^2 \qquad n \geq 1$$

Scanned by CamScanner