L6 - Selection Sort

Thursday, August 20, 2020 10:17 AM

Write a function that finds the smallest element in a list of n integer. The function returns the index of the smallest element in the list.

Find_Min(Arr)

```
let min_Index = 1
for i=2 to Arr.length {
    if Arr[i] < Arr[min_Index]
        min_Index = i
}
return min_Index;
```

Find_Min(Arr)

```
                                                      Operations       WCF       BCF
1.  let min_Index = 1                                     c1             1         1
2.  for i=2 to Arr.length {                               c2             n         n
3.      if Arr[i] < Arr[min_Index]                        c3             n-1       n-1
4.          min_Index = i                                 c4             n-1       0
5.  }
6.  return min_Index;                                     c5             1         1
```

Worst Case Time Complexity
It occurs when the minimum element is at the end of the list and the check in line 3 above succeeds every time.

$T(n) = c1 + c2*n + (c3+c4)*(n-1) + c5$
$= (c2+c3+c4) * n + c1 - (c3+c4) + c5$
$= c10*n + c11 - c12$
$= c10*n + c13$
$= O(n)$ Linear Time Complexity

Best Case Time Complexity
It occurs when the minimum element is at the start of the list and the check in line 3 above fails every time.

$T(n) = c1 + c2*n + c3*(n-1) + c5$
$= (c2+c3) * n + c1 - c3 + c5$
$= O(n)$ Linear Time Complexity $=$ $\Omega(n)$

Since $T(n) = O(n)$ and $T(n) = \Omega(n)$, therefore $T(n) = \theta(n)$

Selection Sort

1. Find the min/max element from the ***current remining*** list
2. Swap the first/last element of the ***current remining*** list with the min/max element found in the step 1 above
3. Repeat the steps 1 & 2 until the list is sorted

	1	2	3	4	5	6
Step 1:	5	2	4	6	1	3
Step 2:	5	2	4	6	3	1
Step 1:	5	2	4	6	3	1
Step 2:	5	3	4	6	2	1
Step 1:	5	3	4	6	2	1
Step 2:	5	6	4	3	2	1
Step 1:	5	6	4	3	2	1
Step 2:	5	6	4	3	2	1
Step 1:	5	6	4	3	2	1
Step 2:	6	5	4	3	2	1

Find_Min_Index(Arr, size)

```
                                          Operations      WCF      BCF
let min_Index = 1                             c1            1        1
for i=2 to Size {                             c2            n        n
    if Arr[i] < Arr[min_Index]                c3            n-1      n-1
        min_Index = i                         c4            n-1      0
}
return min_Index;                             c5            1        1
```

Selection Sort(Arr)

```
                                                               Operations      Freq
for i= 1 to Arr.length-1{                                      c1              n
      let min_Index = Find_Min_Index(Arr, Arr.length - i + 1)  O(n)+c2         n-1
      swap(Arr[Arr.length-i+1], Arr[min_Index])                $O(n^0)$+c3     n-1
}
```

$$\begin{aligned} T(n) &= c1*n + (O(n) + c2)*(n-1) + (O(n^0)+c3)*(n-1) \\ &<= c1*n +(cn+c2)*(n-1) + (c*1 +c3) *(n-1) \\ &= (c1+c2+c*1+c3-c)*n + c*n^2 + \ldots \\ &= c10*n + c*n^2 + c12 \end{aligned}$$

$T(n) = O(n^2) = \Omega(n^2) = \theta(n^2)$