Pseudocode:
Array[j] refers to the light side whereas array[j+1] refers to dark side

Alternating algorithm:

```
Make a temp disk state                                          1 tu
Make a local variable that will function as swap count: swap    1 tu
For i from 0 thru n in array                                    n/2+1 tu

        For j from 0 thru n-1  in array                         n tu
            If array[j] is greater than array[j+1] then         2 tu
                Increment swap                                  2 tu
                Swap array[j] and array[j+1]                    3 tu
Return array, swap
```

Lawnmower algorithm:

```
Make a temp disk state 1 tu
Make a local variable that will function as swap count: swap 1 tu
For i from 0 to n in array                          - n/2 + 1 tu
    For s from total count - 1 downto 1 in array 1-n tu
        If array[s] is less than array[s-1]     - 2 tu
            Increment swap                      - 2 tu
            Swap array[s] and array [s-1]       - 3 tu
    For j from 0 to n-1 in array                - n tu
        If array[j] is greater than array[j+1]  - 2 tu
            Increment swap                      -2 tu
            Swap array[j] and array [j+1]       - 3 tu
```

Analysis:

Alternate algorithm

$T(n) = 1 + 1 + (n/2+1)(n)(2+\max(5, 0)$
$= 2 + (n/2+1)(n)(7)$
$= 2 + (n/2+1)(7n)$
$= 2 + (7n^2)/2 + 7n$

= (7n^2)/2 + 7n + 2
O((7n^2)/2 + 7n + 2)
O((7n^2)/2)
O(n^2)

Lawnmower algorithm
T(n) = 1+1+((n/2)+1) ([(1-n)(2+max(6,0))]+[n(2+max(5,0)])
= 2+((n/2)+1) ([8-8n]+[7n])
= 2+((n/2)+1) (8-1n)
= 2+((8n/2)+1)(8-1n)
= 2+(8n/2)-(n^2/2)+8-n
= 10+(8n/2)-(n^2/2)+8-n
= 10+3n-(n^2/2)
= (n^2/2)-3n-10
O((n^2/2)-3n-10))
O((n^2/2))
O(n^2)