1 + 1 + (n + 2)(2 + (n-1)/2) +1)(6)

3n^2 + 21n + 33

3n^2 + 21n + 33 <= O(n^3)

c= 57, no = 1 => 57 <= 57

Therefore it belongs to O(n^3) by def

Pseudocode

(1) Alternate Algorithm

```
sorted_disks sort_alternate(disk_state beforSwap){
    int numberSwap = 0                                   1
    disk_state state = beforSwap                         1

    for (int i = 0; i < Size + 1; i++)                   n + 2
    {
        if (i % 2 == 0)                                  2
        {                                                (n-1)/2 +1
            for (int currentIndex = 0; currentIndex < Size -1; currentIndex = currentIndex +2)
            {
                if (State.get(currentIndex) != State.get(currentIndex + 1)
                    if(State.get(currentIndex) == DISK_DARK && State.get(currentIndex +1) == DISK_LIGHT)
                    {
                        Swap
                        numberSwap + +
                    }
            }
        }
        else                                             n-2/2 + 1
        {
            for (int currentIndex = 0; currentIndex < Size - 2; currentIndex = currentIndex +2)
            {
                if( state.get(currentIndex) != state.get(currentIndex +1)
                    if(State.get(currentIndex) == DISK_DARK && State.get(currentIndex +1) == DISK_LIGHT)
                    {
                        state.swap(currentIndex)
                        numberSwap + +
                    }
            }
        }
    }
    return sorted_disk(disk_state(state), numberSwap)    1
```

② Lawnmower Algorithm

```
sorted_disks sort_lawnmower(const disk_state& before)
{ int numberSwap = 0                                          1
  disk_state state = before                                   1
  for (int i=0; i < SIZE / 2; i++)                            (n/2) + 1
  { int currentIndex = 0                                      1
    while (currentIndex+1 < Size)                             n + 1
    { if (State.get(currentIndex) != state.get(currentIndex+1))   1
3       { if (state.get(currentIndex) == DISKDARK && state.get(currentIndex+1) == DI
1         { swap(currentIndex)
1           numberSwap++
          }                                                   3 + [((n/2) + 1)* (7n+8)* (7n+7)]
      }
1     currentIndex++
    }
    while (currentIndex > 0)                                  n + 1
1   { if (state.get(currentIndex-1) != state.get(currentIndex))
3       { if (state.get(currentIndex) == DISKDARK && state.get(currentIndex+1) == DI
1         { swap(currentIndex-1)
1           numberSwap++
          }
      }
1     current index--;
1   return
```