left to right

```
sorted_disks surt_left_to right(const disk_state & before) {
  int swapCount = 0;                                          1 ] 1+1=2
  disk_state temporary = before;                              1 ]

  for (size_t i = 0; i < temporary.dosk_count(); i++) {                n/2
      for(size_t j=i; j< temporary.total-count() -1; j++) {           n
          if(temporary.get(j) > temporary.get(j+1)) {          2  ] 5+2=7   ] 7n   ] 7n^2/2
              temporary.swap(j);                               3 ] 3+2=5
              swapCount++;                                     2 ]
          }
      }
  }
   assert(before.is_alternating());                           1 ] 1+2=3
  return sorted-disks(disk_state(temporary), swapCount);      2 ]
}
```

$2+3+\frac{7n^2}{2}$

$\boxed{= 5+\frac{7n^2}{2}}$

```
sorted_disks sort_lawnmover(const disk_state & before) {
  int swapCount = 0;                                          1 ]
  disk_state temporary = before;                              1 ] 3
  bool increment;                                             1 ]
  for(size_t i = 0; i<temporary.dosk_count(); i++) {                  n/2
      increment = (i%2 == 0)? true : false;                           3
      for(size t j = increment ? i: temporary.total-count()-2 ; j < temporary.total_count()
              -1; increment? j++ : j--) {                                      8+3=11   11n/2
          if (temporary.get(j) > temporary.get(j+1)) {         3 ] 3+5=8
              temporary.swap(j);                               3 ] 5
              swapCount++;                                     2 ]
          }
      }
  }
  assert(before.is_alternating());                            1 ] 3
  return Sorted_disks(disk_state(temporary), swapCount);      2 ]
}
```

$\boxed{6+\frac{11n}{2}}$