CSE525 Lec2: Recursion

Stoogesort

```
StoogeSort(A[0 .. n − 1]):
1  if n = 2 and A[0] > A[1]
2        swap A[0] ↔ A[1]
3  else if n > 2
4        m = ⌈2n/3⌉
5        StoogeSort(A[0 .. m − 1])
6        StoogeSort(A[n − m .. n − 1])
7        StoogeSort(A[0 .. m − 1])
```

After 4.	U0	V0	W0
After 5.	U1	V1	W1=W0
After 6.	U2=U1	V2	W2
After 7.	U3	V3	W3=W2

3 partitions of A: U V W

Proof by induction on the length of A.

Induction hypothesis: ???

Prove that: Assuming IH, U3, V3 & W3 are individually sorted.

Prove that: Assuming IH, all x in U3 <= all y in V3 <= all z in W3.

Thm: StoogeSort is correct.

Q: What is its worst-case complexity ?

7,8,9	4,5,6	1,2,3

Prove that: Assuming IH, U3, V3 & W3 are individually sorted.

After line 5, U1 is in increasing order, and so is V1.

After line 6, V2 is in increasing order, and so is W2.

U2 remains in increasing order.

After line 7, U3 is increasing order, and so is V3.

W3 =W2 remains in increasing order.

```
StoogeSort(A[0 .. n − 1]) :
1  if n = 2 and A[0] > A[1]
2      swap A[0] ↔ A[1]
3  else if n > 2
4      m = ⌈2n/3⌉
5      StoogeSort(A[0 .. m − 1])
6      StoogeSort(A[n − m .. n − 1])
7      StoogeSort(A[0 .. m − 1])
```

After 4.	U0	V0	W0
After 5.	U1	V1	W1=W0
After 6.	U2=U1	V2	W2
After 7.	U3	V3	W3=W2

$m = \lceil \frac{2n}{3} \rceil$

U	V	W
0 … n-m-1	n-m … m-1	m … n-1

Prove that: Assuming IH, all x in U3 < all y in V3 < all z in W3.

```
StoogeSort(A[0 .. n − 1]):
1  if n = 2 and A[0] > A[1]
2        swap A[0] ↔ A[1]
3  else if n > 2
4        m = ⌈2n/3⌉
5        StoogeSort(A[0 .. m − 1])
6        StoogeSort(A[n − m .. n − 1])
7        StoogeSort(A[0 .. m − 1])
```

After line 7, since U3 and V3 are sorted, It means all e in U3 < all e' in V3.

If we could prove all e' in V3 < all e'' in W3=W2, we are done.

Each e' in V3 either belongs from U1=U2 or from V2. There can be two cases.

1. If e' belongs to V2, since all elements in V2 < all elements in W2, so, e' < all elements in W2.

2. If e' belongs to U2=U1, then we have few further cases.
   Let minV1 = minimum element in V1. Since e' is in U1, e' < minV1.
   a. If minV1 <= minimum in W2, all elements in W2 > min(W2) => minV1 > e'
   b. If minV1 > min(W2), then all of V1 >= minV1 > min(W2), so W2 should contain all elements of V1 and min(W2). But this is a contradiction since |W2| <= |V1|.

After 4.	U0	V0	W0
After 5.	U1	V1	W1=W0
After 6.	U2=U1	V2	W2
After 7.	U3	V3	W3=W2

Intuition

n=10 m=7 StoogeSort([0 … 6]), StoogeSort([3 … 9]), StoogeSort([0 … 6])
U,V,W: [0 … 2] [3 … 6] [7 … 9]

proof

Generally, U,V,W : [0 … n-m-1] [n-m … m-1] [m … n-1]
|U| = n-m <= n/3, |W| = n-m <= n/3, |V| = n - |U| - |W| >= n/3