Sort1

Sort1(S)

While there are integers x and y in S such that x precedes y in S and x > y

Swap x and y in S

Return S

Is Sort1an algorithm if a selection rule is specified?

Correctness

Correct Algorithm: Well-defined sequence of computational steps that *produce a correct output for every valid input.*

Proof of Correctness

At each stage of Sort1 let $T_j = \Sigma_{k \le j} S_k$

Claim 1: T_j is non-increasing

Claim 2: In each swap some T_j decreases

Sort1 Analysis

	S	T
Input:	5, 3, 8, 1, 2	5, 8, 16, 17, 19
Swap 3 and 2:	5, 2, 8, 1, 3	5, 7, 15, 16, 19
Swap 3 and 5:	3, 2, 8, 1, 5	
Swap 3 and 2:	2, 3, 8, 1, 5	
Swap 2 and 1:	1, 3, 8, 2, 5	
Swap 3 and 2:	1, 2, 8, 3, 5	
Swap 8 and 5:	1, 2, 5, 3, 8	
Swap 5 and 3:	1, 2, 3, 5, 8	

Sort1 Analysis cont.

	S	T
Input:	5, 3, 8, 1, 2	5, 8, 16, 17, 19
Swap 3 and 2:	5, 2, 8, 1, 3	5, 7, 15, 16, 19
Swap 3 and 5:	3, 2, 8, 1, 5	3, 5, 13, 14, 19
Swap 3 and 2:	2, 3, 8, 1, 5	2, 5, 13, 14, 19
Swap 2 and 1:	1, 3, 8, 2, 5	1, 4, 12, 14, 19
Swap 3 and 2:	1, 2, 8, 3, 5	1, 3, 11, 14, 19
Swap 8 and 5:	1, 2, 5, 3, 8	1, 3, 8, 11, 19
Swap 5 and 3:	1, 2, 3, 5, 8	1, 3, 6, 11, 19

Alternative Proof of Correctness

(as suggested in class)

Claim: The number of out-of-order pairs in S decreases with each swap.

At each stage of the algorithm let M(k,m), $1 \le k,m \le n$, be

1 if $k < m$ and $S_k > S_m$

0 otherwise

Note that the number of out-of-order pairs in S is

$\Sigma_{k,m}$ M(k,m)

Alternative Proof cont.

- We can match each pair k,m such that:
  M(k,m) = 0 $\xrightarrow{\text{swap}}$ M(k,m) = 1
- To a pair k',m' such that"
  M(k',m')=1 $\xrightarrow{\text{swap}}$ M(k',m')=0
- And still find an unmatched pair k'',m'' such that"
  M(k'',m'')=1 $\xrightarrow{\text{swap}}$ M(k'',m'')=0

Purely Iterative vs. Recursive Algorithms

- Recursive algorithm: An algorithm which calls itself.
- Purely iterative algorithm: One that doesn't.

```
Sort2(S)
    If ||S||≤1
        Return: S
    Else
        Return: Sort2(S\max-element(S)),max-element(S)
```


Sort2 Example

Input: 5, 3, 8, 1, 2

Sort2 (5,3,8,1,2) = Sort2 (5,3,1,2), 8
= Sort2 (3,1,2), 5, 8
=

Sort2 Example cont.

Input: 5, 3, 8, 1, 2

Sort2 (5,3,8,1,2) = Sort2 (5,3,1,2), 8
= Sort2 (3,1,2), 5, 8
= Sort2 (1,2), 3, 5, 8
= Sort2 (1), 2, 3, 5, 8
= 1, 2, 3, 5, 8

The Questions To Ask:

- Is the algorithm correct?
- Is the algorithm efficient?