CS310: ALGORITHMS AND

DATA STRUCTURES

Selection Sort

- An iterative algorithm that sorts in place
- Finds the smallest element in the collection and places it in the beginning of the collection
- □ After 1st iteration, A[1] is sorted
- \Box After 2nd iteration, A[1...2] is sorted
- \Box After 3rd iteration, A[1...3] is sorted
- □ After n-1th iteration, A[1...n-1] is sorted
- After n-1th iteration, the entire collection is sorted too because the last element is the largest one left

Pseudo Code – Selection Sort

```
For i in 1 to n-1

//find min A[i...n] and place it in A[lower]

tentMin = A[i]

newtentMin = tentMin
```

```
For j = i+1 to n

If A[j] < newtentMin

newtentMin = A[j]

tentMinIndex = j

End If

End For
```

```
If newtentMin != tentMin
swap(A[i], A[tentMinIndex])
End If
End For
```

Quiz: Running Time Expression

```
For i in 1 to n-1

//find min A[i...n] and place it in A[lower]

tentMin = A[i]

newtentMin = tentMin
```

```
For j = i+1 to n

If A[j] < newtentMin

newtentMin = A[j]

tentMinIndex = j

End If

End For
```

```
If newtentMin != tentMin
swap(A[i], A[tentMinIndex])
End If
End For
```

Running Time of Selection Sort

Does not depend on the type of input

Still need to go through all the elements in the array to determine whether the tentative min is the actual min

□ Worst-case T(n) = Average-case T(n) = Best-case T(n)

□ T(n) = K₁(n-1) + K₂
$$\sum_{j=2...n}$$
 (n - j + 1)
= K₁(n-1) + K₂ (n(n+1)-1)/2
= Θ(n²)

Correctness of Iterative Algorithm

Insertion & Selection Sort are iterative algorithms

- Each iteration takes us closer and closer to the required output (i.e. the data structure (array) in the sorted form)
- This assertion about the data structure (array) is called the loop invariant

It can be used to prove that the algorithm is correct

Formal Proof of Correctness

□ Induction Hypothesis: At the end of jth iteration, A[1...j] is sorted

- **Base Case:** At the end of the first iteration, A[1] is sorted
- Induction Step: At the beginning of the iteration, the assertion that A[1...j] is sorted is true, then by induction hypothesis at the end of the (j+1)th iteration, A[1... j+1] is sorted. Because for loop works by moving A[j], A[j-1], A[j-2],A[j-3] and so on by one position to the left until it finds the proper position for A[j+1]. This preserves the loop invariant
- Termination Step: the for loop terminates when j >n. Observing that A[1...n] is the entire array, we conclude the array is sorted. Hence algorithm is correct

Loop Invariants – Specific Descriptions

□ **Insertion Sort:** After ith iteration, A[1...i] is sorted

- Selection Sort: After ith iteration, A[1...i] is sorted with A[1] being the overall smallest element, A[2] the second overall smallest element, etc, all of them being in their final locations
- To prove correctness, use pseudo code to prove that these loop invariants apply through the life of the algorithm – initialization, maintenance, termination