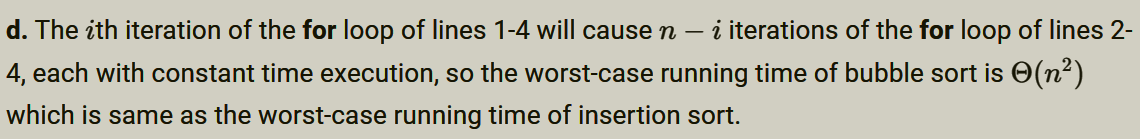
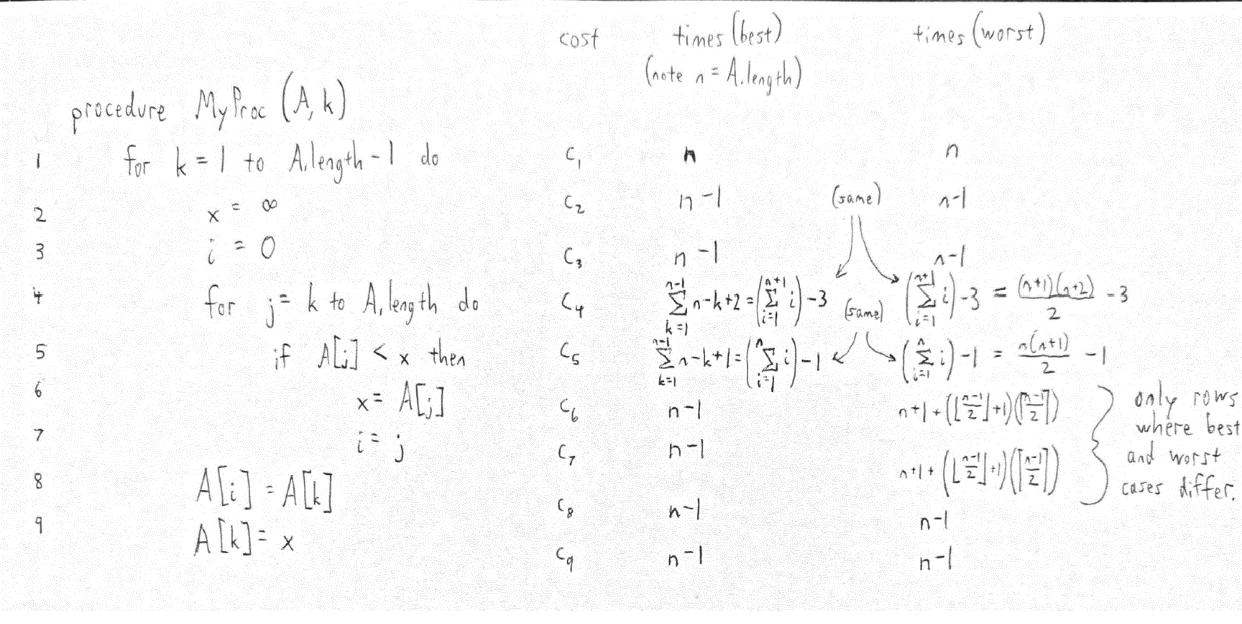
Text, letter

Description automatically generatedText, letter

Description automatically generatedText, letter

Description automatically generatedText, letter

Description automatically generatedA picture containing text

Description automatically generatedText

Description automatically generatedText, letter

Description automatically generatedGraphical user interface, text, application

Description automatically generated

**Loop invariant:**

A contains some permutation of the elements originally in A.

A[1..k-1] contains the k-1 smallest elements of A.

A[1..k-1] is in sorted order.

**Initialization:**

When k=1, no changes to A have been made, and A[1..k-1] contains no elements, so the loop invariant trivially holds.

**Maintenance:**

Assume the loop invariant holds for A[1..k-1]. The inner loop locates the smallest element of A[k..n] and swaps it into location k. Because A[1..k-1] contains the smallest k-1 elements of A, A[k] must now be larger than all A[1..k-1], therefore A[1..k] is now in sorted order. Furthermore, because the smallest element of A[k..n] is now in location k, A[1..k] now contains the smallest k values of A. And because the only changes to A consist of swaps, A still contains some permutation of the elements originally in A.

**Termination:**

The loop stops when k=n. At this point, the loop invariant implies that A[1..n-1] are in sorted order. Because the loop invariant also implies that A[1..n-1] are the smallest n-1 elements in A, A[n] must be the largest; therefore, all A[1..n] are now a sorted permutation of the elements originally in A. Therefore the sorting problem is solved.

Text, letter

Description automatically generatedText

Description automatically generated

Text

Description automatically generated with low confidenceText, letter

Description automatically generatedTable

Description automatically generatedText, letter

Description automatically generated

Text, letter

Description automatically generatedText, letter

Description automatically generatedA close-up of a document

Description automatically generated with medium confidenceText

Description automatically generatedText, letter

Description automatically generatedText, letter

Description automatically generatedText, letter

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedTable

Description automatically generatedText

Description automatically generated

**Loop Invariants**

**Selection Sort**

In the outer loop, array is sorted for first i elements.

In the inner loop, min is always the minimum value in A[i to j].

**Insertion Sort**

loop invariant condition is that the subarray A[0 to i-1] is always sorted.

subarray A[0 to i-1] is a permutation of the original A[0 to i-1]

subarray A[0 to i-1]

**Quick Sort**

Pivot element is placed at its correct position.

Elements less than pivot element lie on the left side of pivot element.

Elements greater than pivot element lie on the right side of pivot element.

**Bubble Sort**

Pivot element is placed at its correct position.

Elements less than pivot element lie on the left side of pivot element.

Elements greater than pivot element lie on the right side of pivot element.

**Find the index of the minimum value in an array of integers**

smallestSoFar in [0,nextToCheck),

and for all k in [0,nextToCheck), A[k] >= A[smallestSoFar].

**Binary search**

if ( A[mid] == a ) then ( start <= mid <= end )