

Due by 2:30pm, Friday, September 21.

For the following problems, state what situation gives the case you are analyzing. For example, you might say that the worst case occurs when the elements are in reverse order. Make all of your analyses as exact as possible. Show your work. If two parts are essentially the same, you can say something like “this part is like Part a.i” and move on. You can also say “we did this in class,” state the answer, and move on.

Problem 1. Assume that you have a sorted list of n elements where you know that exactly one pair of (distinct) elements has been inverted. (An inverted pair of elements do not have to be adjacent.)

- (a) How many comparisons would bubble sort use:
 - (i) In the best case?
 - (ii) In the worst case?
 - (iii) In the average case?
- (b) How many exchanges would bubble sort use:
 - (i) In the best case?
 - (ii) In the worst case?
 - (iii) In the average case?

Problem 2. Repeat Problem 1 for the “modified” bubble sort (where the algorithm ends an iteration at the last location where the previous iteration did an exchange, if any exchanges occurred).

Bubble Sort (Modified)

```

i ← n-1;   new_i ← i
while i > 0 do begin
  for j = 1 to i do begin
    if A[j] > A[j+1] then begin
      A[j] ↔ A[j+1]
      new_i ← j
    end if
  end for
  i ← new_i-1;   new_i ← i
end while

```

Do not worry about average number of comparisons (i.e., do not do part a.iii).

Problem 3. Repeat Problem 1a for insertion sort (using the version of insertion sort with a sentinel). (Count only comparisons, not moves.)

Problem 4. Repeat Problem 1 for selection sort.