634944
For the Insertion sort algorithm; what is its best case and worst case performance?

*a. Best: O(n)

Worst: $O(n^2)$

b. Best: O(n)
Worst: O(n)

c. Best: $O(\log_2 n)$

Worst: $O(n^2)$

d. Best: $O(n^2)$

Worst: $O(n^2)$

e. None of the above.

f. "
g. "
h. "
i. "
j. "

General Feedback:

Insertion sort, if given an already sorted list, will still perform O(n) comparisons to ascertain the list is sorted. If the list is "reverse sorted," then the first pass will require 1 exchange. The second pass will require 2 exchanges, etc. Hence, in the worst case, $O(n^2)$ exchanges.

634947
For the selection sort algorithm; what is its best case and worst case running time?

a. Best: O(1)
Worst: O(n)

b. Best: O(n)

Worst: $O(n^2)$

c. Best: $O(\log_2 n)$

Worst: O(n)

*d. Best: $O(n^2)$

Worst: $O(n^2)$

e. None of the above.

f. "
g. "
h. "
i. "
j. "

General Feedback:

Selection sort repeatedly runs the Find-largest algorithm as its helper function. So, regardless of the list's initial ordering, Find-largest will cost n-1 comparisons for the first pass, n-2 for the second, etc. Hence selection sort's run time performence is independent of the list's initial ordering: $O(n^2)$

633400
You see the expression n = 100000 in some code that successfully compiles. What type can n not be?

a. int

*b. short

c. float

d. double

e. long

f. "
g. "
h. "
i. "
j. "

General Feedback:

Shorts can only hold values in [-32768, 32767].