632266
Suppose you are trying to choose between an array and a singly linked list to store the data in your Java program. Which arguments would correctly support one side or the other?

a. Linked lists are better suited to a program where the amount of data can change unpredictably.

b. Arrays provide more efficient access to individual elements.

c. Linked lists provide more efficient access to individual elements.

*d. A and B only

e. A, B, and C

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

General Feedback:

Array elements can be accessed in constant time, but access to an element of a linked list is O(n), so B is correct and C is incorrect. A is also correct, because arrays require you to guess how much memory will be needed and set aside a fixed amount, while linked lists use memory as needed for the data that are being stored. Since A and B are correct and C is not, the answer is D.

633398
The insertion sort operates by maintaining a sorted list. When a new element is added, we traverse the list sequentially until will find the new element's appropriate location. Suppose instead that the new location was found using a binary search instead of a sequential search. What is the complexity of this new binary insertion sort?

a. O(N)

b. O(log N)

c. O(N + log N)

d. O(N * log N)

*e. $O(N^2)$

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

General Feedback:

The binary search will indeed speed up finding the location of the newly added object. However, we still have the problem of shifting all subsequent elements. This algorithm, like the regular insertion sort, is $O(N^2)$.

633573 2
Suppose you try to perform a binary search on a 5-element array sorted in the reverse order of what the binary search algorithm expects. How many of the items in this array will be found if they are searched for?

a. 5

b. 0

*c. 1

d. 2

e. 3

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

General Feedback:

Only the middle element will be found. The remaining elements will not be contained in the subranges that we narrow our search to.