Algorithmics Screening Exam, Fall 2005

A *growable array* is an indexed list whose size increases when the user attempts to insert an element and the growable array is already full. If the growable array is not yet full, it inserts the new element at the first empty location in its tail. If the growable array is already full, it first allocates a new growable array of some size (that size is determined by the growable array's resize strategy) that is larger than the current one. Then it copies all the elements presently stored in the current growable array to the new growable array, inserts the new element into the new growable array at the first empty location in its tail, deallocates the old growable array, and, finally, replaces the old growable array with the new growable array.

The computational model for the following questions counts **only** element insertions and element copies. An element insertion costs one unit of time. An element copy costs one unit of time as well. Ignore any other possible costs (like, array allocation and deallocation, virtual memory, garbage collection, or synchronization in multi-threaded environments).

In the following questions, growable arrays grow only by a sequence of n insertion requests. You may ignore deletions. Further, n is initially unknown, we want methods that work well regardless of the value of n (when n is large).

1. Assume that a growable array starts at size c and grows by c every time it is full. That is, it grows from c to $2c$, from $2c$ to $3c$, and so on, where $c \geq 1$ is an integer constant. Call this an *additive strategy.*

   (a) What is the size of the largest growable array allocated?

   (b) How many growable arrays are created?

   (c) How many element insertions plus element copyings occur in total?

2. Assume that a growable array starts at size c and grows by a factor of c every time it is full. That is, it grows from c to c^2, from c^2 to c^3, and so on, where $c \geq 2$ is an integer constant. Call this a *multiplicative strategy.*

   (a) What is the size of the largest growable array allocated?

   (b) How many growable arrays are created?

   (c) How many element insertions plus element copyings occur in total?

3. (a) What is the best choice between an additive strategy using $c = 1$ and a multiplicative strategy using $c = 2$?

   (b) What is the best choice between a multiplicative strategy using $c = 2$ and a multiplicative strategy using $c = 3$?

   (c) What can you say about the best choices for c among all additive and multiplicative strategies?