MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001 Structure and Interpretation of Computer Programs
Spring, 2007

Recitation 4b, Wed. February 21

More Order of Growth Problems   Dr. Kimberle Koile

From 6.001 Fall 2005 Quiz 1:

Suppose that you want to sort a list of elements. Here is a procedure for sorting:

```
(define  (find-best best todo compare)
   (if  (null?  todo)
        best
       (if (compare (car todo) best)
           (find-best (car todo) (cdr todo compare)
           (find-best best (cdr todo) compare))))

(define (remove elt todo same)
   (if (null? todo)
      '()
      (if (same elt (car todo))
          (cdr todo)
          (cons (car todo) (remove elt (cdr todo) same)))))

(define (sort data compare same)
   (let ((trial (find-best (car data) (cdr data) compare)))
     (let ((todo (remove trial data same)))
       (if (mull? todo)
           (list trial)
           (cons trial (sort todo compare same))))))
```

To sort a list of numbers called my-list in ascending order, for example, we could write:

```
  (sort my-list (lambda (x y) (<  x  y))  =)
```

For each of the following questions choose the letter corresponding to the option that best describes the order of growth of the process in question.

  A: constant

  B: linear

  C: exponential

  D: quadratic

  E: logarithmic

  F: something else

What is the order of growth in **time** for **find-best**?   B

What is the order of growth in **space** for **find-best**?   A

What is the order of growth in **time** for **remove**?   B

What is the order of growth in **space** for **remove**?   B

What is the order of growth in **time** for **sort**?   D

What is the order of growth in **space** for **sort**?   B