Exam 1

- Handed out at end of Friday’s class, due at the beginning of Monday’s class
- Open non-human resources but no help from other people
- Covers everything through today including:
  - Lectures 1-15, Book Chapters 2-7, PS 1-4
  - Chapter 8 (out today) is not covered (but understanding it may help you prepare for exam)
- Review Session, Weds 6:30 in Olsson 228E

Sorting Cost

\begin{verbatim}
(define (best-first-sort lst cf)
  (if (null? lst) lst
      (let ((best (find-best lst cf)))
        (cons best (best-first-sort (delete lst best) cf))))

(define (find-best lst cf)
  (if (= 1 (length lst)) (car lst)
      (pick-better cf (car lst) (find-best (cdr lst) cf))))
\end{verbatim}

The running time of best-first-sort is in $\Theta(n^2)$ where $n$ is the number of elements in the input list.

Assuming, the procedure passed as cf has constant running time.

Divide and Conquer sorting?

- Best first sort: find the lowest in the list, add it to the front of the result of sorting the list after deleting the lowest
- Insertion sort: insert the first element of the list in the right place in the sorted rest of the list

\begin{verbatim}
(insert-sort lst cf)
  (define (insert-sort lst cf)
    (if (null? lst) null
        (insert-one (car lst)
          (insert-sort (cdr lst) cf) cf)))))

(insert-one el lst cf)
  (define (insert-one el lst cf)
    (if (null? lst) (list el)
        (if (cf el (car lst)) (cons el lst)
            (cons (car lst)
              (insert-one el (cdr lst) cf)))))
\end{verbatim}
Lecture 15: Quicker Sorting

How much work is insert-sort?

\[
\text{(define (insert-sort lst cf)}
\]
\[
\text{  (if (null? lst) null}
\]
\[
\text{    (insert-one (car lst) (insert-sort (cdr lst) cf) cf)))}
\]
\[
\text{(define (insert-one el lst cf)}
\]
\[
\text{  (if (null? lst) (list el)
\]
\[
\text{    (if (cf el (car lst)) (cons el lst)
\]
\[
\text{      (cons (car lst) (insert-one el (cdr lst) cf)))))}
\]

How many times does insert-sort evaluate insert-one?

\(n\) times (once for each element)

insert-sort has running time in \(\Theta(n^2)\) where \(n\) is the number of elements in the input list

Which is better?

- Is insert-sort faster than best-first-sort?

Can we do better?

\[
\text{(insert-one < 88}
\]
\[
\text{  (list 1 2 3 5 6 23 63 77 89 90))}
\]

Suppose we had procedures

- (first-half lst)
- (second-half lst)

that quickly divided the list in two halves?
Charge

- Exam 1 is out Friday, due Monday
- Exam Review, Wednesday 6:30 in Olsson 228E