**Driver.cpp execution:**

#faculty members: 10

contents:

-inf -inf -inf -inf -inf -inf

berger berger berger

cioch

erdly erdly erdly erdly erdly

fukuda

jackels

olson olson olson

stiber

sung

unknown unknown

zander zander

+inf +inf +inf +inf +inf +inf

deleting unknown

#faculty members: 9

contents:

-inf -inf -inf -inf -inf -inf

berger berger berger

cioch

erdly erdly erdly erdly erdly

fukuda

jackels

olson olson olson

stiber

sung

zander zander

+inf +inf +inf +inf +inf +inf

finding stiber = 1

create another list

finding stiber = 1

#faculty members: 9

cost of find = 104

**Performance Results:**

dlist's find cost = 6491439

mtflist's find cost = 66448

translist's find cost = 6422436

skip's find cost = 2078098

**Performance Consideration:**

Ranking the performance results from lowest to highest cost: mtflist, translist, dlist, and skiplist; shows that mtflist costs the least and the skiplist costs the most. If the items in statistics.cpp randomly accessed items in the list the rankings should change. When items are randomly selected the chance of an item being chosen multiple times increases. Since the transpose and mtf lists put the most recently accessed items at the top of the list, they should cost more from accessing the same item multiple times. The cost for skip list shouldn’t increase by that much since the list is sorting and accessing is O(log n).