Brian Engel

Vector Sorting

The code in this project was written to sort a vector of bids. There are two different sorting algorithms that we used, the selection sort and the quick sort. Nothing was too difficult to implement, since there were comments in all the methods showing you exactly what to do. Honestly, the hardest part of the project was getting the existing code into the IDE in the first place. There were definitely a couple of things that I learned coding wise also. I learned how to use a timer in an application, and I have never used the vector swap element function before either.

Pseudocode for selection sort

Selection(numbers)

integer i

integer j

integer smallest

integer temp

for i = 0; i < numbers.size - 1; i = i + 1

smallest = i

for j = i + 1; j < numbers.size; j = j + 1

if (numbers[j] < numbers[smallest])

smallest = j

temp = numbers[i]

numbers[i] = numbers[smallest]

numbers[smallest] = temp

Pseudocode for quick sort

Partition function

Partition(numbers, lowIndex, highIndex)

integer midpoint

numbers array type pivot

boolean done  
 midpoint = lowIndex + (highIndex - lowIndex) / 2  
 pivot = numbers[midpoint]  
   
 done = false  
 while (not done)  
 while (numbers[lowIndex] < pivot)  
 lowIndex = lowIndex + 1  
 while (pivot < numbers[highIndex])  
 highIndex = highIndex - 1

if (lowIndex >= highIndex)  
 done = true  
 else  
 temp = numbers[lowIndex]  
 numbers[lowIndex] = numbers[highIndex]  
 numbers[highIndex] = temp  
 lowIndex = lowIndex + 1  
 highIndex = highIndex - 1  
 return highIndex

Quick sort function

Quicksort(numbers, lowIndex, highIndex)

integer lowEndIndex  
 if (lowIndex >= highIndex)   
 return  
 lowEndIndex = Partition(numbers, lowIndex, highIndex)   
 Quicksort(numbers, lowIndex, lowEndIndex)  
 Quicksort(numbers, lowEndIndex + 1, highIndex)

Main (to run the sorts)

Array numbers

Selection(numbers)

Quicksort(numbers, 0, numbers.size - 1)