

Problem #3

How could you simply modify the recursion in our version of quicksort to allow the program to switch to a simpler routine, say, insertion sort, for subarrays of length below some cutoff value? Write such a program and experiment with the cutoff sizes 4, 8, and 16 for an array of 5000 randomly generated floats to see what the effect on the number of comparisons might be.

In order to implement what the question asks for, we need to add an extra conditional inside our quicksort implementation. In this conditional, we simply check to see if $(\text{last index} - \text{first index} \leq \text{cutoff_size})$. When this condition proves to be true, we call insertion sort on this portion of the array then return.

Original Quicksort:

```
void quickSort(int a[], int first, int last )
{
    int pivotElement;
    if(first < last)
    {
        pivotElement = pivot(a, first, last);
        quickSort(a, first, pivotElement-1);
        quickSort(a, pivotElement+1, last);
    }
}
```

Modified Quicksort:

```
void quickSort( int a[], int first, int last )
{
    int pivotElement;
    if(first < last && (last-first)>CUTOFF)
    {
        pivotElement = pivot(a, first, last);
        quickSort(a, first, pivotElement-1);
        quickSort(a, pivotElement+1, last);
    }
    if ((last-first)<=CUTOFF){
        insertion_sort(a,first,last);
    }
}
```

Results

```
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ g++ number3.cpp
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ ./a.out
Cut Off Length: 4
Number of Comparisons: 73695
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ g++ number3.cpp
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ ./a.out
Cut Off Length: 8
Number of Comparisons: 73386
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ g++ number3.cpp
[Thomass-MacBook-Pro-3:Assignment 2 thomasnguyen$ ./a.out
Cut Off Length: 16
Number of Comparisons: 75423
```