COMP 228 DATA STRUCTURES

PROGRAMMING PROJECT #4, SORTS

Ryan Brooks

**Swap and Comparison Count Results**

|  |  |  |
| --- | --- | --- |
| Sort | Number of Swaps | Number of Comparisons |
| Selection | 49 | 1225 |
| Bubble | 606 | 1225 |
| Better Bubble | 606 | 1222 |
| Insertion | 606 | 654 |
| Merge | 104 | 222 |
| Quick | 70 | 314 |
| Heap | 239 | 528 |

My selection sort amazingly uses O(N^2) to sort the set of numbers that is the same through the whole program. (N-1)(N-2)… N(N-1)/2…2^2/2-N/2… is the selection sort sequence. Bubble sort brings smaller data values to the top of array, each iteration puts the smallest unsorted element into its correct place. Bubble sort is also O(N^2) because they require the same amount of work in terms in the number of comparisons. The better bubble does a type of ”Dirty Flag” tactic where I throw in the “CompCount += 1” right after the for loop, than at the end of the logic I add in my variable to be “False” if “z” isn’t in range of “indexLast”. I see insertion sort as an organizer for sorted linked lists and tree, these types of sorts go by O(N^2) but when an value is in its rightful place there are only comparisons and no swaps in that instance of runtime.

Merge sorts use O(N log\_2 N) which Is faster than selection sorts, bubble sorts and insertion sorts. By Making sure “CompCount” and SwapCount” stayed along with the first half and second half of the array, implementation can stay correct for that instance. The “quicksort” In this program, made “spiltTheRange” useful to look at “first” than “indexSplit – 1” than the reverse idea of “indexSplit +1” than look variable “last”. Heap sorts takes the maximum element of the heap and puts it in its most effective spot of the array, tree, linked list, etc. In this situation of the “heapsort” method, all I needed was “SwapCount” to arrange root element by doing, “SwapCount += 1 swap(0, i) heapify(I, 0)”.

By practicing python code or even code in general is making me realize how algorithms are getting more useful in different languages. Different frameworks and tools are made every so often, which gives me a new challenge to excel at!

Palindromes Test Phrases

1. 1231 FALSE
2. Hey Paul (HeyPaul) FALSE
3. 7653 fsddsf FALSE
4. sharks are in the bay FALSE
5. Table salt is bad FALSE
6. Sadas TRUE
7. Live in bliss,. .,ssilb ni evil (LiveinblissssilbnieviL) TRUE
8. tseb eht uoy ifok kofi you the best (tsebehtuoyifokkofiyouthebest) TRUE
9. 191ihddhi191 TRUE
10. wty33./0033./,ytw// (wty330033ytw) TRUE