COL 216 REPORT

STAGE 3

K LAXMAN

2018CS50408

Objective: using the both stage 1 and stage 2, implementing a recursive merge sort function

In this stage I have implemented recursive merge sort

- In the stage 3, I have used ARMSim 2.0.1 and angel instruction plugin set
- Inputs of my code are ,firstly the user have to enter the size of list
- Next according to the list size input has to be given
- Next input will be comparison mode whether 0 or 1
  - 0 means sensitive comparison mode
  - 1 means insensitive comparison mode
- Next input is duplicate removal
  - 0 means remove the duplicate
  - 1 means don't remove the duplicate
- Output is of format which merge sorts the list according to the input and returns the size of the sorted list also
- Example for working of my code to better explain it clearly

Example of how my ARMSim terminal prompt appears:
Input

3 => size of list expected from user
ball
donkey

cat

0 => here 0 is sensitive

0 =>here 0 is remove duplicate

Ex: ballcatmonkey3 => **output** of mergesort list with size user given

This is the final result of the code which shows the sorted(merge) list from the unsorted list .

The pseudo code of the merge sort function is :

i.e. This is the basic idea which I have implemented

Merge_sort (arr, beg, end)

If beg<end:

Set mid= (beg+end)/2

Merge_sort (arr,beg,mid) //Merge_sort is recursively called here

Merge_sort (arr,mid+1,end)

Merge(arr,beg,mid, end) //here Merge is the stage2 function getting called in my code

End of if

End Merge_sort

- First, I stored the list of pointers that were pointers for the input strings, and then I passed the pointer for the list of pointers as input to the merge sort function.

- I have used usefulfunctions.s , stage1 in this assignment and I have implemented mergesort function in the stage3 file

- My merged list function in stage2 is working good but in stage3,while doing the recursive call in the function for merged sort ,the addresses previously stored are getting overwritten and I am losing the sorted list. I have tried my best to use stack pointer but it is showing so many errors . Other than that all parts are implemented in this stage3

- I have used both comparison mode i.e. sensitive and insensitive and duplicate removal too