ASSIGNMENT 3 DESIGN DOCUMENT

Chris Moon

February 2023

1 Goal

Implement several sorting algorithms in C, and write a test harness allowing the user to run, and collect statistics on each algorithm.

2 Pseudocode

TEST HARNESS

-This is a main function using GETOPT: Allows the user to run and display the various math functions from the terminal

-include all .h files (sorting algorithms, and the set header)

-Specify the command line options: "ahbsqrnp"

-using a switch statement, write cases for each option

-every sorting function will add a value to a set

-for example -s would add 2 to the set

-r sets the seed for the random number generator, used to fill up a list with random numbers to be sorted

-n sets the size of the list to be sorted

-p sets the number of elements to be printed from the list

-p can be at max, the size of the list

-h displays a help message a terminates the program

-h instead displays a help message and returns 1 to terminate the program

-after the switch statement, check the set for values

-if a certain value is in the set (meaning the option has been input into the terminal), run the corresponding sorting algorithm

-the sorting algorithms are run on a randomized array, with memory allocated using malloc

-the array values are randomized using srandom(seed) and random() to get a random value

-write an array scrambling function, looping through the array length and setting a random value

-use this scrambling function between each sorting function, to unsort the array

SET

-a set is defined a 32 bit int

-in order to use the set, define it first

-Set name = empty set (nothing inside, all 0 bits)

-each bit represents a value

-adding a value means setting the bit at the value to 1

-ex: if the set was a 4 bit int, and I wanted to add the value 2, my set would look like 0100

STATS

-a stats struct includes two variables

-moves (whenever an array element is shifted) and compares (whenever 2 array elements are compared)

-in order to use the stats struct and function, define the struct

-Stats name;

-the stats struct is passed into the sorting functions as a parameter, to allow them to track moves and compares

SHELL

-Include the stats header

-write a gap function to be used in the shell sort algorithm:

-if the gap is = 1, then return 0, if the gap = 2 then return 1

-else return 5 times gap / 11

-SHELL ALGORITHM:

-loop from gap function, with the exit condition gap ¿ 0

-increment by setting gap to gap function(gap)

-inside the loop, nest another for loop

-loop from gap, to the number of elements in the list to be sorted

QUICK

-if there are less than 2 elements, do nothing

-move the pivot element in the array

-split array using the pivot (if the element it greater or less)

HEAP

-max child function: returns greater of two children values

-fix heap after moving the largest array value

-build a heap, which is a tree type data structure, where higher values are parents of the tree

BATCHER

-bit length function returns the bit length of a number

-example: bit length of 3 is 2 (11).

-compare function swaps array elements if x ¿ y

MAKEFILE

-Compiles and formats all .c files

-compiles all .c files

-formats c and h files