asgn3 design

Ryan Hui

January 2023

Files

-batcher.c
-batcher.h
-shell.c
-shell.h
-gaps.h
-heap.c
-heap.h
-quick.c
-quick.h
-set.c
-set.h
-stats.c
-stats.h
-sorting.c

Design

Make shell, quick, heap, and batcher sort while tracking stats. Make a program that takes in commands and runs all the sorts while allowing for the user to set the seed, size of array, elements, and prints the program usage.

Pseudocode

Shell Sort

shell sort function
for gap in gaps
for i in range of gap and length of array

j equals i
temp equals array index i

while j greater than equal to gap and temp less than array index j - gap

array index j equals array index j minus gap
j equals j minus gap
array index j equals temp

Heap Sort

max child function
left equals 2 * first
right equals left + 1

if right less than equal to last and array index

right -1 greater than array left -1
return left

fix heap function
found equals false
mother equals first
great equals max child of array, mother, and last

while mother less than or equal to last mod 2

if array index mother - 1 less than array index great - 1

array index mother - 1 equals array index great -1
array index great - 1 equals array index mother - 1
mother equals great
great equals max child of array, mother, and last

else

found equals true

build heap function
for father in range of last mod 2, first -1, and -1

fix heap of array, father and last

heap sort function
first equals 1
last equals length of array
build heap of array, first, and last

```
for leaf in range of last, first, and -1

array index first -1 equals array leaf - 1
array index leaf - 1 equals array first - 1
fix heap of array first and leaf - 1
```

Quick Sort

```
partition function
i equals lo - 1
for j in range of lo and hi

if array index j - 1 less than array index hi - 1

increment i by 1
array index i - 1 equals array index j - 1
array index j - 1 equals array index i - 1

array index i equals array index hi - 1
array index h - 1 equals array index i
return i + 1

quick sorter function
if lo less than hi
p equals partition of array, lo, and hi
quick sorter of array, lo, and p - 1
quick sorter of array, p + 1, and hi

function quick sort
quick sorter of array, 1, and length of array
```

Batcher Sort

```
comparator function
if array index x less than array index y

array index x equals array index y
array index y equals array index x

function batcher sort
if length of array equals 0
return

n equals length of array
```

t equals n bit length
p equals i left shit t - 1

while p greater than 0
for i in range of 0,n and -d

if i and p equals r

comparator of A, i, and i + d
d equals q - p
q right shift equals 1
r equals p

p right shift equals 1

set.c

set empty function
return empty set

set universal function
return a set with every possible member

set insert function
inserts x into s

set remove function
function removes x from s

set member function
returns bool given x value in s

set union function
returns union of two sets

set intersect function
returns intersection of two sets

set difference function
returns difference of two sets

set complement
function returns complement of a given set

sorting.c

main function
get opt
-a: runs all sorts
-s: enables shell sort
-b: enables batcher sort
-q: enables quicksort
-h: enables heap sort
-r: enables random seed
-n: sets array size
-p: prints elements
-H: enables program usage -?: error enables program usage
-default: default enables program usage

if print h

print program usage
end program

if shell

set random seed
loop through size of and fill array with random numbers
call shell sort function
print columns and stats

if print size less than elements

x equals size
else
x equals elements

loop through x
if elements exceed 5 or last element
print elements in column with new line

reset stats

if batcher

set random seed
loop through size of and fill array with random numbers
call batcher sort function
print columns and stats

```
if print size less than elements

x equals size
else
x equals elements

loop through x
if elements exceed 5 or last element
print elements in column with new line

reset stats

if heap

set random seed
loop through size of and fill array with random numbers
call heap sort function
print columns and stats

if print size less than elements

x equals size
else
x equals elements

loop through x
if elements exceed 5 or last element
print elements in column with new line

reset stats

if quick

set random seed
loop through size of and fill array with random numbers
call quick sort function
print columns and stats

if print size less than elements

x equals size
else
x equals elements

loop through x
if elements exceed 5 or last element
```

print elements in column with new line

reset stats

Pseudocode Structure

-each sort uses functions from stat.c. these functions used are move which tracks a move done to the array and counts for 1, compare which compares two array elements and counts for 1, and swap which swaps the elements values which counts for 3.

-shell sort function takes in an array and sorts it by sorting pairs of elements that are separated by gaps which will be continuously reduced until the array is sorted. uses gap.h header file.

-batcher sort takes in an array and divides like a sorting network. the comparator function will compare values traveling along the wires and sort them until the array is sorted

-heap sort takes in an array and builds a heap tree and sets a max element with the max child function The fix heap function places the max child at the end of the array with children nodes equal to the value of the parent. the fix heap function is repeated until the array is sorted.

-quick sort takes in an array and the partition function divides it into two arrays based off a pivot value. it then sorts the two arrays until the array is sorted using a recursive quick sorter function.

-sorting.c uses a get opt for the user to enable the sorts and settings of the program. arrays are populated with random numbers depending on the seed and sorted by the chosen sort. array elements are printed in of columns of 5 with their sort and stats above it. program usage is displayed when -H or an invalid command is entered.

-set.c uses bitwise set operators to achieve the function's goals.

Credit

-cse13s discord
-piazza