PA02 - Kth Smallest Value

Generated by Doxygen 1.8.11

Contents

1 File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

2 File Documentation

2.1 CS302/Projects/PA02/PA02.cpp File Reference

This file contains the code for Programming Assignment 2, finding the kth smallest value in a set of data.

```
#include <iostream>
#include <fstream>
#include <string>
```

Functions

- void ReadInData (ifstream *sentFile, int *arrayStart)
  This function is used to read in the data from the given data file.
- int GetDataAmount (ifstream *sentFile)
  This function is used to scan through the data file to see the amount of values within the file.
- void PrintValues (int *arrayStart, int size)
  This function is used to print every value within the array.
- void PrintValues (int *arrayStart, int from, int to, int pivotIndex)
  This function is used to print the data within a given scope of the values array.
- int FindKthValue (int kth, int *arrayStart, int firstPos, int lastPos)
  This function is used to find the kth smallest value within an array of values.
- void OutputToLog (int kth, int firstPos, int lastPos, int pivotIndex, int pivot)
  This function is used to print various values that are important for debugging the partitioning algorithm.
- void OutputToLog (string sentString)
  This function simply prints a given string to the output file.
- int **main** (int argc, char *argv[])

Variables

- ofstream **logFile**

2.1.1 Detailed Description

This file contains the code for Programming Assignment 2, finding the kth smallest value in a set of data.

Author

Bryce Monaco (base file by Shehryar Khattak)

This file contains the code for all functions used in the PA02 assignment. The user inputs a value k and the program returns the kth smallest value in given data.

Version

1.0

Note

The program assumes that the data file is named "data02.txt", if it is not a new file name can be given as a command line argument

2.1.2 Function Documentation

2.1.2.1 int FindKthValue (int *kth,* int * *arrayStart,* int *firstPos,* int *lastPos*)

This function is used to find the kth smallest value within an array of values.

This function is used to find the kth smalles value with in an array of values. It recursively calls itself and sends a partition of the previously scanned array until it finds the value.

Algorithm The main algorithm of this function is the partitioning algorithm which puts values less than the pivot value to the left (left partition) and value

Parameters

in	*kth*	The value to search for.
in	*arrayStart*	The pointer to the start of the array to store the values in.
in	*firstPos*	The index value of the array to start the scope at.
in	*lastPos*	The index value of the array to end the scope at.
out	*pivot*	Returns pivot if it is the kth smallest value in the array

Returns

The value of pivot is returned if it is the kth smalles value within the given array, otherwise the function makes a recursive call.

Note

None.

2.1.2.2 int GetDataAmount (ifstream * *sentFile*)

This function is used to scan through the data file to see the amount of values within the file.

The function continuously reads through the data file to count the number of values within it. The number found is used to dynamically size the values array in main().

Algorithm Continuously reads through the file checking if it has reached the end of the file. It increments an integer to keep count of the values and retu

Parameters

in	*sentFile*	The data file opened in main(), contains the data to be read in.
out	*count*	The number of values within the data file.

Returns

Returns the amount of valid values within the data file.

Note

None.

2.1.2.3 void OutputToLog (int *kth,* int *firstPos,* int *lastPos,* int *pivotIndex,* int *pivot*)

This function is used to print various values that are important for debugging the partitioning algorithm.

This function simply formats and prints the values sent to it from the current run of the FindKthValue function.

Parameters

in	*kth*	The value to search for.
in	*firstPos*	The index value of the array to start the scope at.
in	*lastPos*	The index value of the array to end the scope at.
in	*pivotIndex*	The index value of the pivot, used to print which value is the pivot.
in	*pivot*	The value of the pivot.
out	*None.*	

Returns

None.

Note

None.

2.1.2.4 void OutputToLog (string *sentString*)

This function simply prints a given string to the output file.

This function simply prints a given string to the output file and then moves to the next line.

Parameters

in	*sentString*	The string to be printed
out	*None.*	

Returns

None.

Note

None.

2.1.2.5 void PrintValues (int * *arrayStart,* int *size*)

This function is used to print every value within the array.

This function starts at the address pointed to by arrayStart and prints values until zie amount of values have been printed.

Algorithm Prints each value within the array pointed to by arrayStart.

Parameters

in	*arrayStart*	The pointer to the start of the array to store the values in.
in	*size*	The amount of values within the array, used to prevent the arrayTrav pointer from going out of bounds.
out	*None.*	

Returns

None.

Note

If size is incorrect then the entire array might not be printed or the arrayTrav will run out of bounds. Output is formatted as (index number) value.

2.1.2.6 void PrintValues (int * *arrayStart,* int *from,* int *to,* int *pivotIndex*)

This function is used to print the data within a given scope of the values array.

denotes that the value is the pivot.

This function this function starts at the value within the array at index from and prints all values up to and including the value at index to. Used for printing partitions.

Algorithm Moves the arrayTrav to the start of the scope, then prints all values starting from arrayStart[from] and ending with arrayStart[to].

Parameters

in	*arrayStart*	The pointer to the start of the array to store the values in.
in	*from*	The array index value of where to start printing.
in	*to*	The array index value of the end of the scope.
in	*pivotIndex*	The index value of the pivot, used to print which value is the pivot.
out	*None.*	

Returns

None.

Note

Output is formatted as (index number) value.

to the right of a value denotes that it is the pivot.

2.1.2.7 void ReadInData (ifstream * *sentFile,* int * *arrayStart*)

This function is used to read in the data from the given data file.

The function continuously reads in data line by line until it reaches the end of the file. It stores the values in the array pointed to by arrayStart.

Algorithm Continuously reads through the file checking if it has reached the end of the file. It reads in a new value and stores it in the current memory p

Parameters

in	*sentFile*	The data file opened in main(), contains the data to be read in.
in	*arrayStart*	The pointer to the start of the array to store the values in.
out	*None.*	

Returns

None.

Note

None.

Index