CSCI 301 Computer Science II Summer 2023

Project 3 – Recursive Solutions

Due Date: 11:50 pm on July 06

**Introduction:**

The goal of this program is to find the kth smallest value in an array of elements. For the function, the array parameter is passed as a pointer and When testing your functions in the main program, the array that is to be passed to the function must be generated as a dynamic array using a new operator. In this implementation, the partition function is responsible for partitioning the array around the pivot and returning the index of the pivot element. It uses the Lomuto partition scheme.

**Code list**:

#include <iostream>

/\*\*

\* problem description: Implement the algorithm kSmall, discussed in the class (Chapter 2, Section 2.4.4) by a C+

+ function. Use the first value of the array as the pivot. The partition of the array must be

implemented using a function as well. Test your functions in a main program

\*

\* Name: Algassimou Diallo

\* startID: el8524jv

\* Instructor: Jie Meichsner

\* Due date: 07/06/2023

\*/

/\*\*

\* Partition the array around the pivot and return the index of the pivot element

\*

\* @param arr The array to be partitioned

\* @param low The starting index of the partition

\* @param high The ending index of the partition

\* @return The index of the pivot element

\*/

int partition(int\* arr, int low, int high) {

int pivot = arr[low]; // Choose the first element as the pivot

int i = low + 1;

int j = high;

while (i <= j) {

if (arr[i] < pivot && arr[j] > pivot) {

std::swap(arr[i], arr[j]);

i++;

j--;

} else if (arr[i] >= pivot) {

i++;

} else if (arr[j] <= pivot) {

j--;

}

}

std::swap(arr[low], arr[j]); // Move the pivot element to its correct position

return j;

}

/\*\*

\* Find the kth smallest element in the array using the kSmall algorithm

\*

\* @param arr The array of integers

\* @param size The size of the array

\* @param k The value of k for the kth smallest element

\* @return The kth smallest element

\*/

int kSmall(int\* arr, int size, int k) {

int low = 0;

int high = size - 1;

while (low <= high) {

int pivotIndex = partition(arr, low, high);

if (pivotIndex == k - 1) {

return arr[pivotIndex]; // Found the kth smallest element

} else if (pivotIndex > k - 1) {

high = pivotIndex - 1; // Adjust the partition range for the left side of the pivot

} else {

low = pivotIndex + 1; // Adjust the partition range for the right side of the pivot

}

}

// If k is out of range or array is empty

return -1;

}

int main() {

int size;

int k;

std::cout << "Enter the size of the array: ";

std::cin >> size;

if (size <= 0) {

std::cout << "Invalid array size." << std::endl;

return 0;

}

int\* arr = new int[size]; // Dynamic array allocation

std::cout << "Enter the elements of the array: ";

for (int i = 0; i < size; i++) {

std::cin >> arr[i];

}

std::cout << "Enter the value of k for the kth smallest element: ";

std::cin >> k;

if (k <= 0 || k > size) {

std::cout << "Invalid value of k." << std::endl;

delete[] arr; // Release memory before exiting

return 0;

}

int kthSmallest = kSmall(arr, size, k);

if (kthSmallest != -1) {

std::cout << "The kth smallest element is: " << kthSmallest << std::endl;

} else {

std::cout << "Invalid value of k or array is empty." << std::endl;

}

delete[] arr; // Release memory

return 0;

}

**User Document:**

The kSmall function finds the kth smallest element in the array using the kSmall algorithm. It repeatedly partitions the array until the kth smallest element is found.

In the main program, the user is prompted to enter the size of the array and its elements. The dynamic array is allocated using the new operator. The user is also prompted to enter the value of k for the kth smallest element. The kSmall function is called to find the kth smallest element, and the result is displayed.

Finally, the dynamically allocated array memory is released using the delete operator to avoid memory leaks.

The file of this project is located in the centOS folder: el8524jv/csci301/project3

To compile this program please run the command: g++ kthsmallest.cpp then ./a.out

Test cases:

| Tests | Input | Output |
| --- | --- | --- |
| kth small is in S1 | Array size = 10  Array[ ]= 3 4 7 5 6 9 0 3 4 5;  K = 4 | Kth smallest: 5 |
| kth small is the pivot | Array size =10  Array[ ]=1 2 3 4 5 7 9 4 0  k=5 | Kth smallest: 4 |
| kth small is in S2 | array size = 7  Array [ ]=1 2 3 4 5 6 7  k=3 | Kth smallest: 5 |
| Boundary values | Array size = 4  Array[ ] = 3 4 6 3 4 6 7 | Out of boundary: Invalid value of k. |
| K value boundary | Array size = 4  Array[ ]= 2 0 3 4 k= 5 | Invalid value of k. |

The script tests from centos:

* Invalid \_K:

Script started on Thu 06 Jul 2023 09:42:26 PM CDT

]0;el8524jv@csci4:~/csci301/project3[?1034h[el8524jv@csci4 project3]$ ./a.out

Enter the size of the array: 4

Enter the elements of the array: 2 0 4 5

Enter the value of k for the kth smallest element: 5

Invalid value of k.

]0;el8524jv@csci4:~/csci301/project3[el8524jv@csci4 project3]$ exit

Script done on Thu 06 Jul 2023 09:43:05 PM CDT

* Array is out of boundary:

Script started on Thu 06 Jul 2023 09:39:32 PM CDT

]0;el8524jv@csci4:~/csci301/project3[?1034h[el8524jv@csci4 project3]$ ./a.out

Enter the size of the array: 4

Enter the elements of the array: 3 4 6 3 4 6 7

Enter the value of k for the kth smallest element: The kth smallest element is: 3

]0;el8524jv@csci4:~/csci301/project3[el8524jv@csci4 project3]$ exit

Script done on Thu 06 Jul 2023 09:41:09 PM CDT

* Kth smallest is the pivot:

Script started on Thu 06 Jul 2023 09:39:32 PM CDT

]0;el8524jv@csci4:~/csci301/project3[?1034h[el8524jv@csci4 project3]$ ./a.out

Enter the size of the array: 4

Enter the elements of the array: 3 4 6 3 4 6 7

Enter the value of k for the kth smallest element: The kth smallest element is: 3

]0;el8524jv@csci4:~/csci301/project3[el8524jv@csci4 project3]$ exit

Script done on Thu 06 Jul 2023 09:41:09 PM CDT

* Kth smallest is in S1:

Script started on Thu 06 Jul 2023 09:27:18 PM CDT

]0;el8524jv@csci4:~/csci301/project3[?1034h[el8524jv@csci4 project3]$ ./a.out

Enter the size of the array: 10

Enter the elements of the array: 3 4 7 5 6 9 0 3 4 5

Enter the value of k for the kth smallest element: 4

The kth smallest element is: 5

]0;el8524jv@csci4:~/csci301/project3[el8524jv@csci4 project3]$ exit

Script done on Thu 06 Jul 2023 09:28:22 PM CDT

* Kth smallest is in s2:

Script started on Thu 06 Jul 2023 09:37:38 PM CDT

]0;el8524jv@csci4:~/csci301/project3[?1034h[el8524jv@csci4 project3]$ .a[K/a.out

Enter the size of the array: 7

Enter the elements of the array: 1 2 3 4 5 6 7

Enter the value of k for the kth smallest element: 3

The kth smallest element is: 5

]0;el8524jv@csci4:~/csci301/project3[el8524jv@csci4 project3]$ exit

Script done on Thu 06 Jul 2023 09:38:12 PM CDT

**Summary:**

This program was successfully built from a local IDE and then transferred to centOS and tested multiple times. We registered no failure during our testing. All the tests were successful. We did some boundary testing to measure the error ranger.