

1. Введение

- 1) Текстовая формулировка задачи
- 2) Пример кода, решающего данную задачу
- 3) Скриншот программы

2. Ход работы

2.1. Код приложения

Сортировки Быстрая и Слиянием

1)

```
#include <iostream>
```

```
using namespace std;
```

```
void quickSort(int arr[], int left, int right) {
```

```
    int i = left, j = right;
```

```
    int temp;
```

```
    int pivot = arr[(left + right) / 2];
```

```
    while (i <= j) {
```

```
        while (arr[i] < pivot)
```

```
            i++;
```

```
        while (arr[j] > pivot)
```

```
            j--;
```

```
        if (i <= j) {
```

```
            temp = arr[i];
```

```
            arr[i] = arr[j];
```

```
            arr[j] = temp;
```

```
            i++;
```

```
            j--;
```

```
        }
```

```
    }
```

```
    if (left < j)
```

```
        quickSort(arr, left, j);
```

```
    if (i < right)
```

```
        quickSort(arr, i, right);
```

```
}
```

```
int main() {
```

```
    int arr[] = { 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 };
```

```

    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    quickSort(arr, 0, n - 1);

    cout << "\nSorted array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
2)
#include <iostream>
using namespace std;

void merge(int arr[], int left, int middle, int right) {
    int i, j, k;
    int n1 = middle - left + 1;
    int n2 = right - middle;
    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[middle + 1 + j];

    i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
    }
}

```

```

        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int middle = left + (right - left) / 2;
        mergeSort(arr, left, middle);
        mergeSort(arr, middle + 1, right);
        merge(arr, left, middle, right);
    }
}

int main() {
    int arr[] = { 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 };
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Original array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    mergeSort(arr, 0, n - 1);

    cout << "\nSorted array: ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}

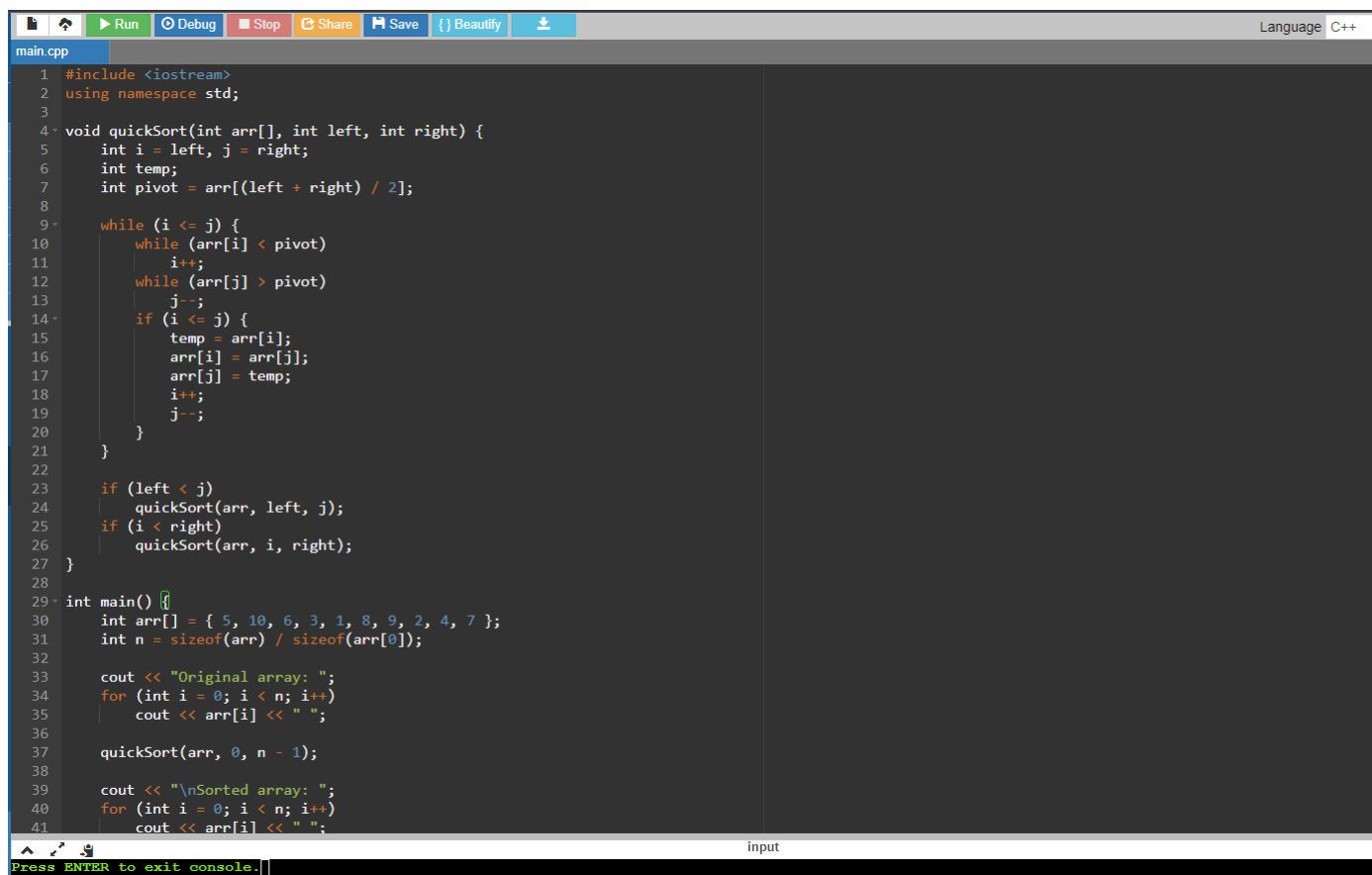
```

2.2. формулы

Общая формула сортировки быстрой и слиянием quickSort(arr, 0, n - 1) и mergeSort(arr, 0, n - 1);

3. Пример скриншота программы

Сортировка быстрая



The screenshot shows a C++ IDE with a dark theme. The top toolbar includes buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C++. The code in main.cpp implements a quicksort algorithm. It includes `<iostream>` and uses the `std` namespace. The `quickSort` function is recursive, taking an array and left/right indices. It finds a pivot at the middle, partitions the array by moving elements greater than the pivot to the right, and then recursively sorts the left and right sub-arrays. The `main` function initializes an array `{ 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 }`, prints it, calls `quickSort`, and prints the sorted array. The console at the bottom shows the prompt "Press ENTER to exit console." and the word "input" on the right.

```
1 #include <iostream>
2 using namespace std;
3
4 void quickSort(int arr[], int left, int right) {
5     int i = left, j = right;
6     int temp;
7     int pivot = arr[(left + right) / 2];
8
9     while (i <= j) {
10         while (arr[i] < pivot)
11             i++;
12         while (arr[j] > pivot)
13             j--;
14         if (i <= j) {
15             temp = arr[i];
16             arr[i] = arr[j];
17             arr[j] = temp;
18             i++;
19             j--;
20         }
21     }
22
23     if (left < j)
24         quickSort(arr, left, j);
25     if (i < right)
26         quickSort(arr, i, right);
27 }
28
29 int main() {
30     int arr[] = { 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 };
31     int n = sizeof(arr) / sizeof(arr[0]);
32
33     cout << "Original array: ";
34     for (int i = 0; i < n; i++)
35         cout << arr[i] << " ";
36
37     quickSort(arr, 0, n - 1);
38
39     cout << "\nSorted array: ";
40     for (int i = 0; i < n; i++)
41         cout << arr[i] << " ";
42 }
```

Рис. 1. скриншот программы

The screenshot shows a C++ IDE with a file named `main.cpp`. The code implements a quicksort algorithm. The `main` function initializes an array `arr` with the values `{ 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 }` and calls `quickSort(arr, 0, n - 1)`. The program prints the original array and the sorted array.

```
13     j--;  
14     if (i <= j) {  
15         temp = arr[i];  
16         arr[i] = arr[j];  
17         arr[j] = temp;  
18         i++;  
19         j--;  
20     }  
21 }  
22  
23 if (left < j)  
24     quickSort(arr, left, j);  
25 if (i < right)  
26     quickSort(arr, i, right);  
27 }  
28  
29 int main() {  
30     int arr[] = { 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 };  
31     int n = sizeof(arr) / sizeof(arr[0]);  
32  
33     cout << "Original array: ";  
34     for (int i = 0; i < n; i++)  
35         cout << arr[i] << " ";  
36  
37     quickSort(arr, 0, n - 1);  
38  
39     cout << "\nSorted array: ";  
40     for (int i = 0; i < n; i++)  
41         cout << arr[i] << " ";  
42  
43     return 0;  
44 }
```

The output window shows the following text:

```
Original array: 5 10 6 3 1 8 9 2 4 7  
Sorted array: 1 2 3 4 5 6 7 8 9 10  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

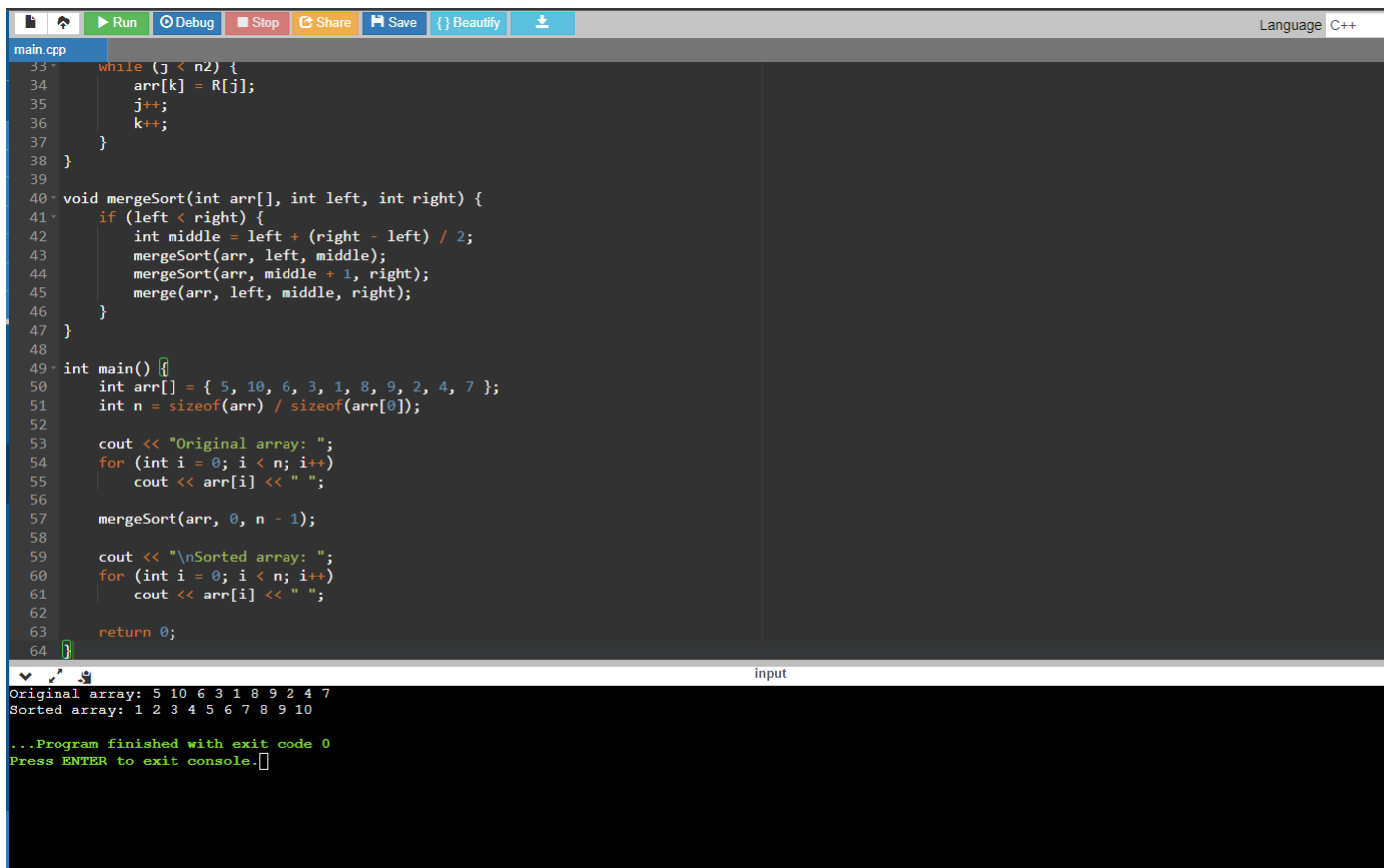
Рис. 2. скриншот программы

Сортировка слиянием

A screenshot of a C++ IDE window titled 'main.cpp'. The window has a toolbar with icons for Run, Debug, Stop, Share, Save, and Beautify. The code is for a merge sort algorithm. It includes the <iostream> header and uses the std namespace. The merge function takes an array and indices for left, middle, and right. It splits the array into two halves, sorts them, and then merges them back. The mergeSort function is also shown, which calls the merge function recursively. The code is written in C++ and is syntax-highlighted.

```
1 #include <iostream>
2 using namespace std;
3
4 void merge(int arr[], int left, int middle, int right) {
5     int i, j, k;
6     int n1 = middle - left + 1;
7     int n2 = right - middle;
8     int L[n1], R[n2];
9
10    for (i = 0; i < n1; i++)
11        L[i] = arr[left + i];
12    for (j = 0; j < n2; j++)
13        R[j] = arr[middle + 1 + j];
14
15    i = 0, j = 0, k = left;
16    while (i < n1 && j < n2) {
17        if (L[i] <= R[j]) {
18            arr[k] = L[i];
19            i++;
20        } else {
21            arr[k] = R[j];
22            j++;
23        }
24        k++;
25    }
26
27    while (i < n1) {
28        arr[k] = L[i];
29        i++;
30        k++;
31    }
32
33    while (j < n2) {
34        arr[k] = R[j];
35        j++;
36        k++;
37    }
38 }
39
40 void mergeSort(int arr[], int left, int right) {
41     if (left < right) {
```

Рис. 3. скриншот программы



```
main.cpp
33 while (j < n2) {
34     arr[k] = R[j];
35     j++;
36     k++;
37 }
38 }
39
40 void mergeSort(int arr[], int left, int right) {
41     if (left < right) {
42         int middle = left + (right - left) / 2;
43         mergeSort(arr, left, middle);
44         mergeSort(arr, middle + 1, right);
45         merge(arr, left, middle, right);
46     }
47 }
48
49 int main() {
50     int arr[] = { 5, 10, 6, 3, 1, 8, 9, 2, 4, 7 };
51     int n = sizeof(arr) / sizeof(arr[0]);
52
53     cout << "Original array: ";
54     for (int i = 0; i < n; i++)
55         cout << arr[i] << " ";
56
57     mergeSort(arr, 0, n - 1);
58
59     cout << "\nSorted array: ";
60     for (int i = 0; i < n; i++)
61         cout << arr[i] << " ";
62
63     return 0;
64 }
```

input

Original array: 5 10 6 3 1 8 9 2 4 7
Sorted array: 1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.

Рис. 4. скриншот программы

4. библиографические ссылки

Для изучения «внутренностей» \TeX необходимо изучить [1], а для использования \LaTeX лучше почитать [2, 3].

Список литературы

- [1] Кнут Д.Э. Всё про \TeX . — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе \LaTeX . — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. \LaTeX в примерах. 2005 г.