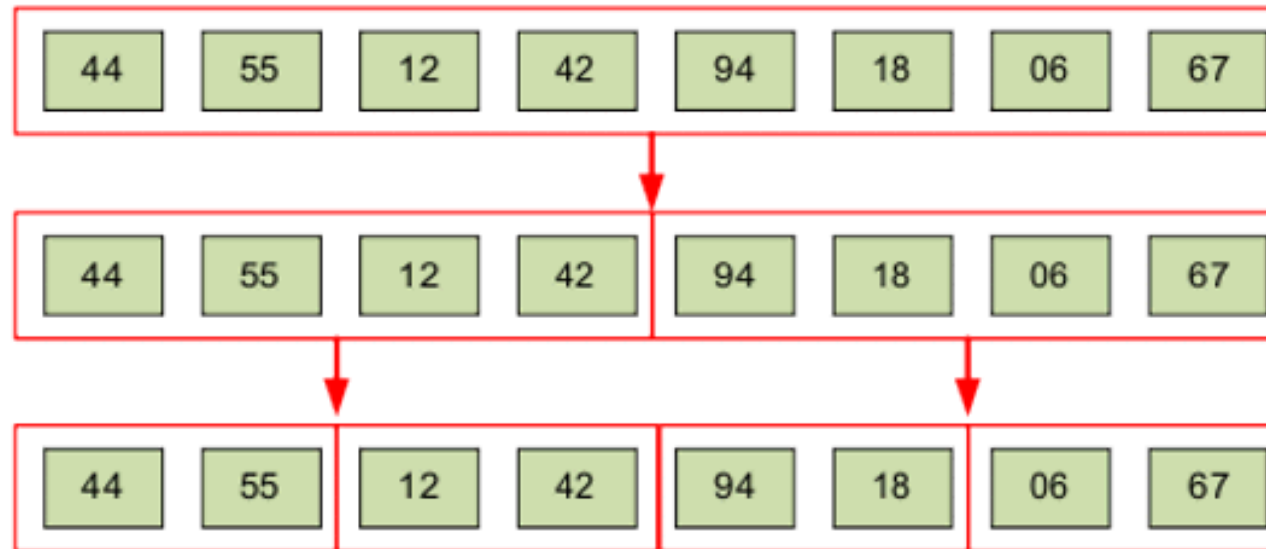


Сортировка нисходящим слиянием

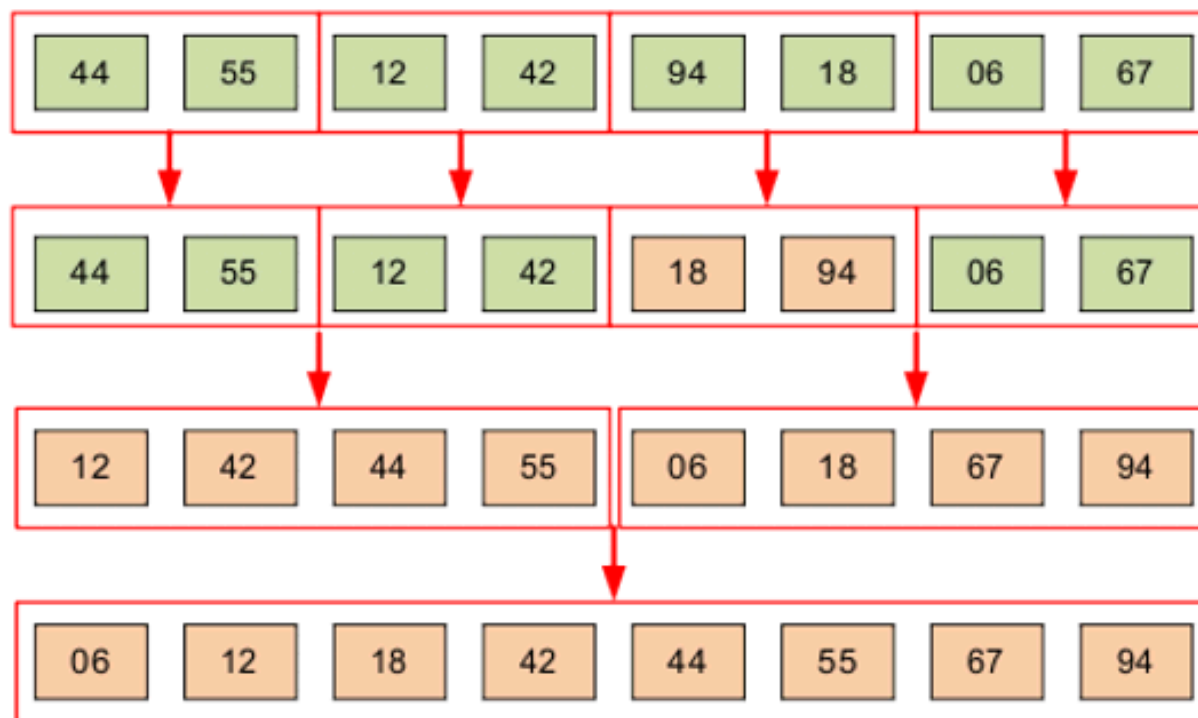
ПОДГОТОВИЛ БРЯНЦЕВ В.А.

Как она работает?

1. Исходная последовательность разбивается на половины, пока не получим подпоследовательности по 2 элемента.



2. Каждую подпоследовательность упорядочиванием методом слияния и получаем готовую отсортированную последовательность.



```

#include <iostream>

using namespace std;

void mergeSort(int* A, int left, int right)
{
    if (left == right) //если границы сомкнулись, останавливаем
    {
        return;
    }
    int middle = (left + right) / 2; //определяем середину
    mergeSort(A, left, middle); //вызываем для левой половины
    mergeSort(A, middle + 1, right); //вызываем для правой половины

    int* tmp = new int[right]; //дополнительный массив
    int first = left; //начало первого пути
    int second = middle + 1; //начало второго пути

    for (int i = 0; i < right - left + 1; i++) //для всех элементов
        //дополнительного массива
        {
            //будем записывать в дополнительный массив меньший из
            //элементов двух путей
            //или остаток первого пути если second > right
            if ((second > right) || ((first <= middle) && (A[first] <
A[second])))
            {
                tmp[i] = A[first];
                first++;
            }
            else
            {
                tmp[i] = A[second];
                second++;
            }
        }

    //переписываем сформированную последовательность в исходный массив
    for (int i = 0; i < right - left + 1; i++)
    {
        A[left + i] = tmp[i];
    }
}

int main()
{
    const int N = 15;
    int A[N];
    for (int i = 0; i < N; i++)
    {
        cin >> A[i];
    }
    mergeSort(A, 0, N - 1);
    for (int i = 0; i < N; i++)

```

```
    {  
        cout << A[i] << " ";  
    }  
}
```