public static String[] leaveShort(String a)

{

String[] s = a.split(" ");

int l = s.length;

String[] str = new String[l/3];

for(int i=0; i<l; i+=3)

{

int min = i;

if(s[i+1].length()<s[i].length())

{

min = i+1;

}

if(s[i+2].length()<s[min].length())

{

min = i+2;

}

str[i/3]=s[min];

}

return str;

}

The time complexity of my algorithm is O(N). I used for loop to iterate every group of three strings to compare their length and leave the shortest one. The i in the for loop is pushed forward by three since every group of three is tested, and the next index will start the next group of three. The for loop has a time complexity of O(N), and other parts are O(1), so the entire algorithm has a time complexity of O(N).