

```
static int Func1(KeyValuePair<int, string>[] a, int low, int high, int key)
{
    int middle = low + ((high - low)/2);

    if (low == high)
        return low;

    if (key > a[middle].Key)
        return Func1(a, middle + 1, high, key);

    return Func1(a, low, middle, key);
}
```

Func1 – функция рекурсивного бинарного поиска в массиве кортежей. На вход функции поступает сам массив «**a**» (причём массив должен быть отсортирован), два числа (**low** и **high**), которые задают интервал внутри которого будет происходить поиск, и **key** – значение которое нужно найти.

На выходе получаем индекс элемента, значение которого либо равно **key** (в случае, если таких элементов несколько, вернёт индекс первого), либо ближайший справа к значению **key**.

Func1 для этого массива при параметрах **low** = 0, **high** = 5, **key** = 2 вернёт 1.

0		1		2		3		4		5	
1	string	2	string	2	string	2	string	4	string	6	string

Func1 для этого массива при параметрах **low** = 0, **high** = 5, **key** = 5 вернёт 5.

0		1		2		3		4		5	
1	string	2	string	2	string	2	string	4	string	6	string

Данный код, по моему мнению, можно немного улучшить:

```
if (key > a[middle].Key)
    return Func1(a, middle + 1, high, key);
```

Заменить на:

```
if (key >= a[middle].Key)
    return Func1(a, middle + 1, high, key);
```

Пусть будет следующая последовательность ключей:

a = {0, 2, 2, 2, 2, 2, ... , 2, 5}

Допустим, происходит вызов **Func1(a, 0, a.Length-1, 2)**. В результате получим индекс первой двойки в последовательности двоек. В **Func2** массив расширится на один элемент и начнется смещение элементов вправо от полученного индекса (первой двойки в последовательности двоек) до конца, что может занять много времени. Поэтому, на мой взгляд, рациональнее было бы получать индекс следующего элемента после последней двойки в последовательности двоек. Достигается это за счёт изменения следующей строки:

```
if (key >= a[middle].Key)
```

В этом случае время на смещение элементов уменьшится.

```

static void Func2(ref KeyValuePair<int, string>[] a, int key, string value)
{
    int pos;
    KeyValuePair<int, string> keyValuePair;

    if (a.Length == 0)
    {
        Array.Resize(ref a, 1);
        keyValuePair = new KeyValuePair<int, string>(key, value);
        a[0] = keyValuePair;
        return;
    }

    if (key < a[0].Key)
        pos = 0;
    else if (key > a[a.Length - 1].Key)
        pos = a.Length;
    else
        pos = Func1(a, 0, a.Length - 1, key);

    Array.Resize(ref a, a.Length + 1);
    for (int i = a.Length - 1; i > pos; i--)
        a[i] = a[i - 1];

    keyValuePair = new KeyValuePair<int, string>(key, value);
    a[pos] = keyValuePair;
}

```

Func2 – функция добавления кортежа в массив кортежей. При чём добавление происходит так, что все значения ключей кортежей массива располагаются в порядке возрастания.

В случае, если массив пустой – просто добавляем в него новый кортеж.

Если в массиве присутствуют другие кортежи, то выполняем проверки:

1. Если значение **key** добавляемого кортежа меньше, чем значение **key** первого кортежа массива, то расширяем массив на один элемент, смещаем все элементы вправо и добавляем наш кортеж в самое начало.
2. Если значение **key** добавляемого кортежа больше, чем значение **key** последнего кортежа массива, то расширяем массив на один элемент, и добавляем наш кортеж в самый конец.
3. С помощью **Func1** находим индекс в массиве, в который нужно записать наш кортеж.

Продемонстрирую это на примере. Имеется массив типа **KeyValuePair** состоящий из 3 элементов.

0		1		2	
2	string	3	string	5	string

1 случай: необходимо добавить в массив кортеж со значением **key** = 1.

Имеем:

0		1		2	
2	string	3	string	5	string

Значение **key** добавляемого кортежа меньше, чем значение **key** первого кортежа массива, поэтому расширяем массив на один элемент:

0		1		2		3	
2	string	3	string	5	string		

Смещаем все элементы массива вправо:

0		1		2		3	
2	string	2	string	3	string	5	string

массива

В первую ячейку записываем наш кортеж:

0		1		2		3	
1	string	2	string	3	string	5	string

2 случай: необходимо добавить в массив кортеж со значением **key** = 8.

Имеем:

0		1		2	
2	string	3	string	5	string

Значение **key** добавляемого кортежа больше, чем значение **key** последнего кортежа массива, поэтому расширяем массив на один элемент:

0		1		2		3	
2	string	3	string	5	string		

И в последнюю ячейку массива записываем наш кортеж:

0		1		2		3	
2	string	3	string	5	string	8	string

3 случай: необходимо добавить в массив кортеж со значением **key** = 4.

Имеем:

0		1		2	
2	string	3	string	5	string

Значение **key** добавляемого кортежа больше, чем значение **key** первого кортежа массива, но меньше, чем значение **key** последнего кортежа массива, поэтому с помощью Func1 находим индекс в массиве, в который нужно записать наш кортеж. Для нашего случая индекс равен 2. Расширяем массив на один элемент:

0		1		2		3	
2	string	3	string	5	string		

Относительно полученного индекса смещаем все элементы вправо:

0		1		2		3	
2	string	3	string	5	string	5	string

И записываем наш кортеж в полученный индекс:

0		1		2		3	
2	string	3	string	4	string	5	string

Улучшение которое можно внести в приведенный код – переименовать функции и дать им осмысленное название в соответствии с их назначением. Также, поиск реализован через рекурсию. Рекурсия использует стек ограниченной глубины. Таким образом, при вызове на массиве с большим кол-вом элементов, функция поиска индекса вообще может не отработать. Здесь либо нужно ставить ограничение на длину входного массива, либо переписывать поиск через цикл.