static int Func1(KeyValuePair<int, string>[] a, int low, int high, int key) /\*словарь a здесь не нужен\*/

{

  int middle = low + ((high - low)/2); // число между low и high

  if (low == high) /\* т.е. рекурсия прервется, когда диапазон уменьшится максимально\*/

    return low;

  if (key > a[middle].Key)

/\* находит значение в словаре по ключу middle и возвращает соответствующий ключ для

этого значения, то есть тот же middle. \*/

    return Func1(a, middle + 1, high, key);

  return Func1(a, low, middle, key);

}

В функции Func1 представлен бинарный (двоичный) поиск - определение значения элемента в середине структуры данных. Полученное значение сравнивается с ключом. Если ключ меньше значения середины, то поиск осуществляется в первой половине элементов, иначе — во второй. Поиск сводится к тому, что вновь определяется значение серединного элемента в выбранной половине и сравнивается с ключом. Процесс продолжается до тех пор, пока не будет найден элемент со значением ключа или не станет пустым интервал для поиска.

Оптимизированный код может выглядеть так:

static int Func1(int low, int high, int key)

{

  int middle = low + ((high - low)/2);

  if (low == high)

    return low;

if (key > middle)

    return Func1(middle + 1, high, key);

  return Func1(low, middle, key);

}

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

{

  int pos;

  KeyValuePair<int, string> keyValuePair; /\* можно обойтись и без этой переменной\*/

  if (a.Length == 0)

  {

    Array.Resize(ref a, 1);

    keyValuePair = new KeyValuePair<int, string>(key, value); /\* тоже можно убрать\*/

    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); /\* изменить под функцию Func1\*/

  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); /\* тоже можно убрать\*/

  a[pos] = keyValuePair;

}

В функции Func2 в передаваемый словарь добавляется новая пара ключ-значение.

Оптимизированный код может выглядеть так:

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

{

  if (a.Length == 0) // если словарь пуст

  {

    Array.Resize(ref a, 1); // увеличиваем размер на 1

    a[0] = new KeyValuePair<int, string>(key, value); // добавляем новую пару

    return;

  }

int pos; // позиция, на которую помещается пара

  if (key < 0) // если ключ отрицательный, то добавление происходит в начало

    pos = 0;

  else if (key > a.Length - 1) /\* если ключ выходит за диапазон словаря, то добавление в конец\*/

    pos = a.Length;

  else

    pos = Func1(0, a.Length - 1, key); // поиск ключа

  Array.Resize(ref a, a.Length + 1);

  for (int i = a.Length - 1; i > pos; i--) /\* смещение элементов на 1 индекс вправо до отметки pos\*/

    a[i] = a[i - 1];

  a[pos] = new KeyValuePair<int, string>(key, value); // добавление новой пары

}