

### 算法 RadixSort

// 基数排序算法,  $p$  为关键词的位数,即每一个关键词  $K_j$  ( $1 \leq j \leq n$ ) 由如下  
//的  $p$  个关键词组成  $key[0], \dots, key[p-1]$ ,  $radix$  为基数

```
void RadixSort(Element *list, const int n, const int p, const int radix)
{
    int *start = new int[radix];
    int *end = new int[radix]; //指向表示桶的每个队列的对头和队尾
    for (int j = 1; j < n; j++) list[j].link = j + 1;
    list[n].link = 0; // 建立初始链表
    int current = 1;
    for (int i = p - 1; i >= 0; i--) // 从低位开始,依次按子关键词  $key[i]$  排序
    {
        for (j = 0; j < radix; j++) start[j] = 0; // 初始化队列表头
        while (current)
        {
            int k = list[current].keys[i];
            if (!start[k]) start[k] = current;
            else list[end[k]].link = current;
            end[k] = current;
            current = list[current].link;
        }
        for (j = 0; start[j] == 0; j++); // 寻找第一个非空队列
        current = start[j];
        int last = end[j];
        for (int k = j + 1; k < radix; k++) // 所有子队列合并为一个大的队列
        {
            if (start[k])
            {
                list[last].link = start[k];
                last = end[k];
            }
        }
        list[last].link = 0;
    }
    delete []start;
    delete []end;
}
```