{第三题 设计快速排序的非递归算法}

PROC QuickSort(VAR R, S:ARRAY[1..n] OF datatype; l,p: integer);

{l是数组起始下标元素，p是结束下标元素, S 是栈}

BEGIN

IF(l >= p) THEN exit;

mid = Partition(R, l, p);

IF(mid > l + 1) THEN [push(R[l]); push(R[mid - 1]);]

IF(mid < p - 1) THEN [push(R[mid + 1]); push(R[p]);

WHILE(S不为空) [

right = pop();

left = pop();

mid = Partition(R, left, right);

IF(left < mid - 1) THEN [push(R[left]); push(R[mid - 1])];

IF(right > mid + 1) THEN [push(R[mid + 1]); push(R[right])];

]

END;

FUNCTION Partition(R:ARRAY[1..n] OF datatype; low, high: integer);

{low, high, mid 分别是存储给定数据数组的数据的下标值}

BEGIN

mid = low + (high - low)/2;

IF(R[low] > R[high]) THEN swap(R[low], R[high]); {交换两者的值}

IF(R[mid] > R[high]) THEN swap(R[mid], R[high]);

IF(R[low] > R[mid]) THEN swap(R[low], R[mid]);

{将low 和 high 之间的元素分为两部分进行排序, x 是选取的轴心元素}

x = R[low];

WHILE(low < high) [

WHILE(low < high) AND (R[high] >= pivot) DO high := high - 1;

R[low] := R[high];

WHILE(low < high) AND (R[low] <= pivot) DO low := low + 1;

R[high] := R[low];

high := high - 1;

]

R[low] := x;

return low;

END;