1190201215 冯开来

问题：在O（n）的时间内，对0到n\*n – 1之间的n个整数进行排序。

将n个数转换为n进制的二位数，再将这些二位数进行两遍基数排序。最后再输出结果

时间复杂度为O（d \* （n+r））= O（2 \* （n+n））= O（4n）

代码

void Radix\_Sort(int figure, Queue A, int n)

{ //figure =2

Queue Q[n];

int data, r;

for (int pass=0; pass<figure; pass++)

{

for (int i=0;i<n;i++)

MakeNull(Q[i]);

while(!Empty(A))

{

data = Front(A);

Dequeue(A);

r = Radix(data, key, pass, n);

Enqueue(data, Q[i]);

}

for (int j=0;j<n;j++)

{

while(!Empty(Q[i]))

{

data = Front(Q[i]);

Dequeue(Q[i]);

Enqueue(data, A);

}

}

}

}

int Radix (int k, int p, int n)

{

int power = 1;

for (int i=0;i<p;i++)

power \*= n;

return (k%(power\*n))/power;

}