**分治法实现循环赛问题**

**代码：**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

void merge(int n, int a[200][100])

{

int m = n / 2;

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

{

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

{

a[i][j + m] = a[i][j] + m; //由左上角的小块算出右上角小块的值

a[i + m][j] = a[i][j + m]; //由右上角的小块算出左下角小块的值

a[i + m][j + m] = a[i][j]; //由左下角的小块算出右下角小块的值

}

}

}

void arrangement(int n, int a[100][100])

{

if (n == 1)

{

a[0][0] = 1;

return;

}

arrangement(n / 2, a);

merge(n, a);

}

int main()

{

int n;

printf("选手的个数为2^k (小于100),请输入k的值：\n");

scanf\_s("%d", &n);

int num[100][100] = { 0 };

int power = pow(2, n);

arrangement(power, num);

printf("这%d个选手的比赛日程表：\n",power);

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

{

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

printf("%d ", num[i][j]);

printf("\n");

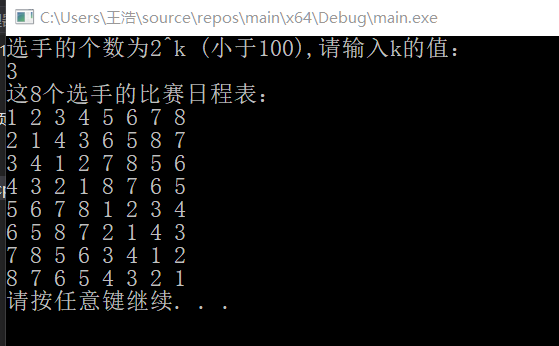
}

system("pause");

return 0;

}

**结果：**



**减治法实现可变规模问题**

**代码：**

#include<stdio.h>

#include<stdlib.h>

int quicksort(int a[], int left, int right, int result, int n)

{

int count = 1;

int i, j;

int temp;

do

{

i = left;

j = right;

temp = a[left];

do

{

//从右向左找到第1个不小于标准值的位置j

while (a[j] > temp && i < j)

j--;

if (i < j)

{

a[i] = a[j];

i++;

}

//从左向右找到第1个不大于标准值的位置i

while (a[i] < temp && i < j)

i++;

if (i < j)

{

a[j] = a[i];

j--;

}

} while (i < j);

a[i] = temp; //将标准值放入它的最终位置

printf("第%d轮排序：", count);

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

printf("%d ", a[k]);

printf("\n");

count++;

if (i == result)

break;

if (i < result)

left = i + 1;

else

right = i - 1;

} while (i != result);

return a[result];

}

int main()

{

int n, m;

int result = 0;

printf("请输入数组元素个数：\n");

scanf\_s("%d", &n);

int a[100] = { 0 };

printf("请输入想要查询的元素：\n");

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

scanf\_s("%d", &a[i]);

printf("请输入想要查询的倒数第几元素：\n");

scanf\_s("%d", &m);

result = quicksort(a, 0, n - 1, m - 1, n);

printf("\n%d是第%d小的元素\n", result, m);

system("pause");

return 0;

}

**结果：**

