1. n个元素正序o(n)，反序o（n^2），数据相同o（n）
2. 无关，对

4.n=1时o（log2n）

6.首先选取堆排序，其次选取快速排序，最后选取二路归并排序

二路归并排序

不选取快速排序

1. 不能，必须采用稳定的排序方法，用栈代替队列，使排序过程不稳定。

11.void DBubbleSort(RecType *R*[],int *n*)

{

int i=0,j;

bool exchange=true;

while (exchange)

{

exchange=false;

for (j=*n*-i-1;j>i;j--)

if (*R*[j].key<*R*[j-1].key)

{

exchange=true;

swap(*R*[j],*R*[j-1]);

}

for (j=i;j<*n*-i-1;j++)

if (*R*[j].key>*R*[j+1].key)

{

exchange=true;

swap(*R*[j],*R*[j+1]);

}

if (!exchange) return;

i++;

}

}

13.KeyType QuickSelect(RecType *R*[],int *s*,int *t*,int *k*)

{

int i=*s*,

j=*t*;

RecType tmp;

if (*s*<*t*)

{

tmp=*R*[*s*];

while (i!=j)

{

while (j>i && *R*[j].key>=tmp.key)

j--;

*R*[i]=*R*[j];

while (i<j && *R*[i].key<=tmp.key)

i++;

*R*[j]=*R*[i];

}

*R*[i]=tmp;

if (*k*-1==i) return *R*[i].key;

else if (*k*-1<i) return QuickSelect(*R*,*s*,i-1,*k*);

else return QuickSelect(*R*,i+1,*t*,*k*);

}

else if (s==t && s==k-1)

return R[k-1].key;

else

return -1;

}

O（n）

14.#include "seqlist.cpp"

#include <malloc.h>

#define Max 3

typedef struct node

{

RecType Rec;

struct node \*next;

}

NodeType;

void RadixSort1(RecType *R*[],int *n*)

{

NodeType\*head[Max],\*tail[Max],\*p,\*t;

int i,k;

for (i=0;i<Max;i++)

head[i]=tail[i]=NULL;

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

{

p=(NodeType \*)malloc(sizeof(NodeType));

p->Rec=*R*[i];

p->next=NULL;

k=*R*[i].key;

if (head[k]==NULL)

{

head[k]=p;

tail[k]=p;

}

else

{

tail[k]->next=p;

tail[k]=p;

}

}

p=NULL;

for (i=0;i<Max;i++)

if (head[i]!=NULL)

{

if (p==NULL)

{

p=head[i];

t=tail[i];

}

else

{

t->next=head[i];

t=tail[i];

}

}

i=0;

while (p!=NULL)

{

*R*[i++]=p->Rec;

p=p->next;

}

}

int main()

{

int i,n=5;

RecType R[MAXL]={{1,'A'},{0,'B'},{0,'C'},{2,'D'},{1,'F'}};printf("排序前:\n");

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

printf("[%d,%c] ",R[i].key,R[i].data);

printf("\n");

RadixSort1(R,n);

printf("排序后:\n");

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

printf("[%d,%c] ",R[i].key,R[i].data);

printf("\n");

return 1;

}

排序前:[1,A][0,B][0,C][2,D][1,F]排序后:[0,B][0,C][1,A][1,F][2,D]

复杂度O（n）