2014年

2019年11月14日 16:54

数据结构部分

一.

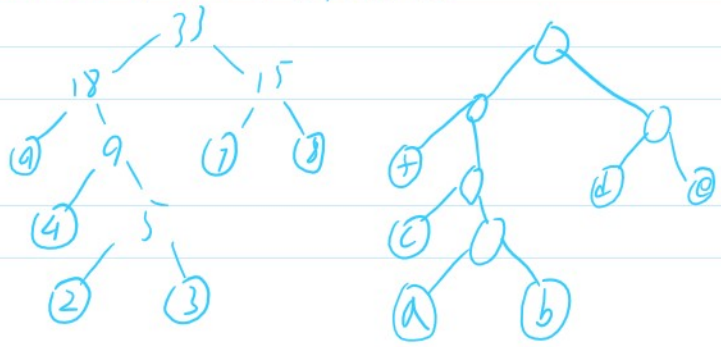
1. C
2. B
3. B
4. A
5. C
6. A
7. A
8. B
9. C
10. A

二.

1.

是图似乎印错了，不做了..

2.



3.

希尔排序:

第1次: 23 9 39 5 33 12 62 48 68

第2次: 23 5 33 9 39 12 62 48 68

第3次: 5 9 12 23 33 39 48 62 68

快速排序:

第1次: 12 9 5 23 68 39 62 48 33

第2次: 5 9 12 23 33 39 62 48 68

第3次: 5 9 12 23 33 39 62 48 68

第4次: 5 9 12 23 33 39 62 48 68

第5次: 5 9 12 23 31 39 48 62 68

三.

1.

```
#include<iostream>
#include<stdlib.h>
using namespace std;
struct List
{
	int data;
	List *next;
};

List* Create()
{
	int tem;
	List *head=(List*)malloc(sizeof(List));
	List *temNode1=head;
	List *temNode2,*temNode3;
	while(cin>>tem,tem!=-1)//输入-1结束
	{

		temNode1->data=tem;
		temNode2=(List*)malloc(sizeof(List));
		temNode3=temNode1;
		temNode1->next=temNode2;
		temNode1=temNode2;
	}
	temNode3->next=NULL;

	return head;
}


void PrintList(List *a)
{
	while(a)
	{
		cout<<a->data<<" ";
		a=a->next;
	}
	cout<<endl;
}
List* MinToHead(List *L)
{
	List *min=L,*tar,*head=L;
	while(L->next)
	{
		if(min->data>L->next->data)
		{
```

```
List* MinToHead(List *L)
{
    List *min=L,*tar,*head=L;
    while(L->next)
    {
        if(min->data>L->next->data)
        {
            min=L->next;
            tar=L;
        }
        L=L->next;
    }
    tar->next=min->next;
    min->next=head;
    return min;
}
int main()
{
    List *a=Create();
    PrintList(a);
    PrintList(MinToHead(a));
}
```

2.

```
#include<iostream>
#include<stdlib.h>
using namespace std;

struct BTnode
{
    int data;
    BTnode *left;
    BTnode *right;
};
bool IsEqual(BTnode *a,BTnode *b)
{
    bool flag=true;
    if(a==NULL&&b==NULL)
        return true;
    else if(a==NULL&&b!=NULL)
        return false;
    else if(a!=NULL&&b==NULL)
        return false;
    else
    {
        if(a->data!=b->data)
            return false;
        if(IsEqual(a->left,b->left)==false)
            return false;
        if(IsEqual(a->right,b->right)==false)
            return false;
    }
    return true;
}

int main()
{
    BTnode *node25=(BTnode*)malloc(sizeof(BTnode));    node25->left=NULL;              node25->
    right=NULL;                node25->data=5;
    BTnode *node24=(BTnode*)malloc(sizeof(BTnode));    node24->left=NULL;              node24->
    right=NULL;                node24->data=4;
    BTnode *node23=(BTnode*)malloc(sizeof(BTnode));    node23->left=NULL;              node23->
    right=NULL;                node23->data=3;
    BTnode *node22=(BTnode*)malloc(sizeof(BTnode));    node22->left=node24;  node22->
    right=node25;  node22->data=2;
    BTnode *node21=(BTnode*)malloc(sizeof(BTnode));    node21->left=node22;  node21->
    right=node23;  node21->data=1;

    BTnode *node17=(BTnode*)malloc(sizeof(BTnode));    node17->left=NULL;              node17->
    right=NULL;                node17->data=7;
    BTnode *node16=(BTnode*)malloc(sizeof(BTnode));    node16->left=NULL;              node16->
    right=NULL;                node16->data=6;
    BTnode *node15=(BTnode*)malloc(sizeof(BTnode));    node15->left=NULL;              node15->
    right=NULL;                node15->data=5;
    BTnode *node14=(BTnode*)malloc(sizeof(BTnode));    node14->left=node16;  node14->
    right=node17;  node14->data=4;
    BTnode *node13=(BTnode*)malloc(sizeof(BTnode));    node13->left=NULL;              node13->
    right=NULL;                node13->data=3;
    BTnode *node12=(BTnode*)malloc(sizeof(BTnode));    node12->left=node14;  node12->
    right=node15;  node12->data=2;
    BTnode *node11=(BTnode*)malloc(sizeof(BTnode));    node11->left=node12;  node11->
    right=node13;  node11->data=1;
    if(IsEqual(node11,node11))
        cout<<"equivalent";
    else
        cout<<"not equivalent";
}
```

数据库部分

一、

1. A
2. C
3. B
4. B
5. D

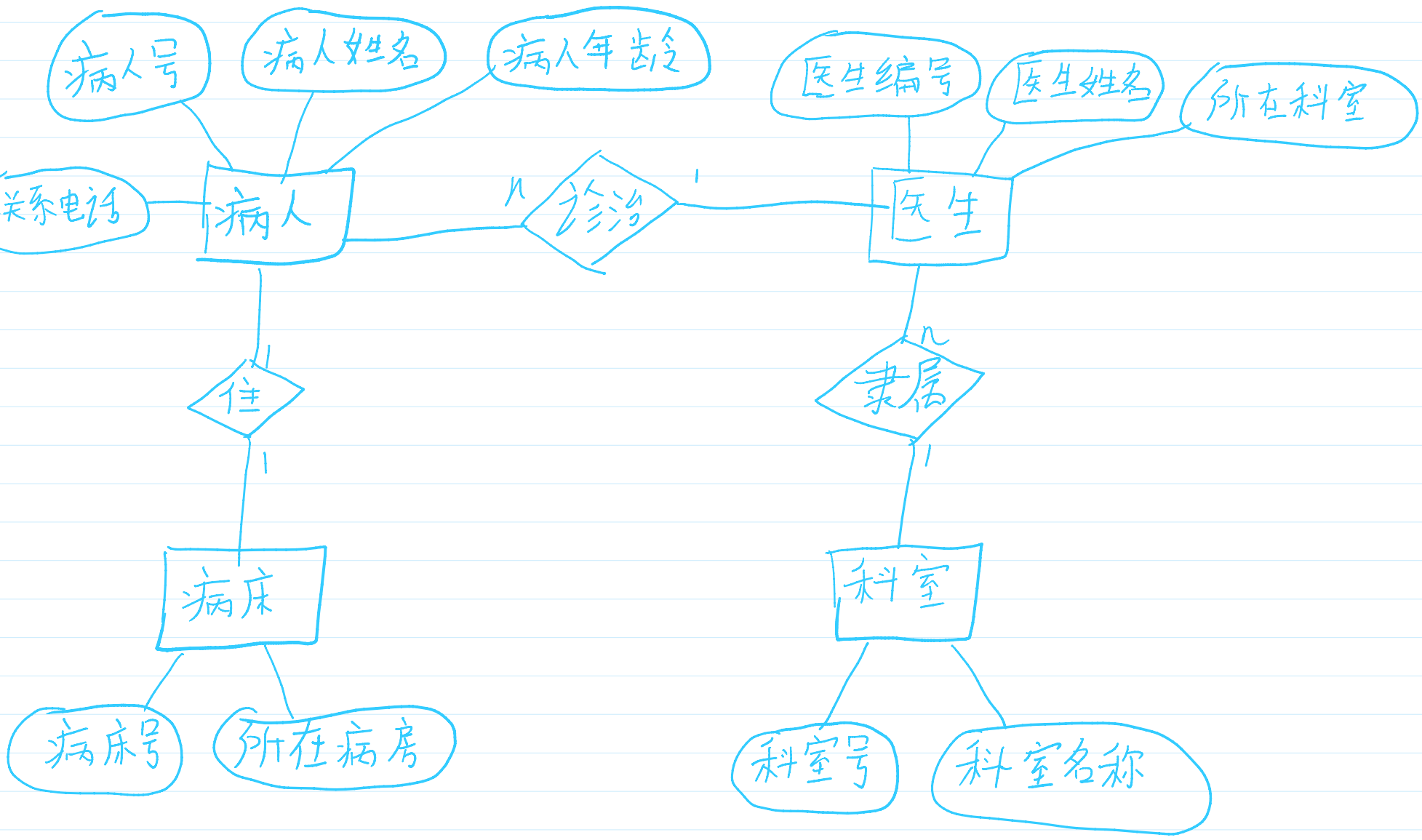
二、

1. $\Pi_{S\#,SNAME}(\sigma_{TNAME='张三'\wedge CNAME='C语言程序'\wedge GRADE>90}(S\bowtie SC\bowtie C\bowtie T))$

2. Select S#,SNAME from S where S# in
   (Select S#,GRADE from SC where GRADE>90 AND
   C# in
   (select CNAME,C# from C where CNAME='C语言程序'
   AND T# in
   (select T# from T where TNAME='张三')))

一、

1.



2.

病人（病人号，病人姓名，病人年龄，联系电话，医生编号）

医生（医生编号，医生姓名，所在科室，科室号）

病床（病人号，病床号，所在病房）

科室（科室号，科室名称）