Ch¬ng 7

S¾p xÕp vµ t×m kiÕm (sorting and searching)

7.1- §Æt bµi to¸n

S¾p xÕp lµ qu¸ tr×nh bè trÝ l¹i c¸c phÇn tö cña mét tËp ®èi tîng nµo ®ã theo mét thø tù Ên ®Þnh t¨ng dÇn (increasing), hoÆc gi¶m dÇn (decreasing). Bµi to¸n s¾p xÕp xuÊt hiÖn trong bÊt kú lÜnh vùc nµo cña tin häc, phôc vô nh÷ng øng dông ri²ng cña hÖ thèng, tõ nh÷ng øng dông È n b²n trong cña HÖ ®iÒu hµnh nh bµi to¸n ®iÒu khiÓn qu¸ tr×nh (Proccess Control Problem), bµi to¸n lËp lÞch cho CPU (CPU Schedulling), bµi to¸n qu¶n lý bé nhí (Memory Management) . . . cho tíi nh÷ng øng dông th«ng th-ênh nh s¾p xÕp d·y sè, s¾p xÕp c¸c tõ, c¸c c©u, c¸c b¶n ghi theo thø tù ®Òu cã li²n quan tíi qu¸ tr×nh s¾p xÕp.

TËp ®èi tîng cÇn ®îc s¾p xÕp cã thÓ xuÊt hiÖn díi nhiÒu d¹ng kh¸c nhau, c¸c ®èi tîng ®ã cã thÓ lµ c¸c ®èi tîng d÷ liÖu kiÓu c¬ b¶n nh s¾p xÕp d·y sè, s¾p xÕp kÝ tù, s¾p xÕp string hoÆc lµ c¸c ®èi tîng tæng qu¸t nh mét cÊu tróc bao gåm mét sè trêng th«ng tin ph¶n ¸nh ®èi tîng. Chóng ta qui íc ®èi tîng cÇn ®îc s¾p xÕp lµ c¸c cÊu tróc, vµ qu¸ tr×nh s¾p xÕp ®îc thùc hiÖn tr²n mét trêng nµo ®ã gäi lµ trêng kho¸.

Cã nhiÒu thuËt to¸n s¾p xÕp kh¸c nhau ®Ó s¾p xÕp c¸c ®èi tîng. Tuy nhi²n, ®Ó lùa chän mét thuËt to¸n s¾p xÕp tèt, chóng ta cÇn ®¸nh gi¸ thuËt to¸n theo c¸c hai khÝa c¹nh: ®ã lµ sù chiÕm dông bé nhí khi ¸p dông gi¶i thuËt vµ thêi gian thùc hiÖn gi¶i thuËt. §èi víi thêi gian thùc hiÖn gi¶i thuËt, chóng ta còng cÇn ®¸nh gi¸ chi phÝ thêi gian trong trêng hîp tèt nhÊt, trung b×nh vµ xÊu nhÊt ®èi víi nguån d÷ liÖu vµo. Chóng ta còng chØ ®a ra nh÷ng kü thuËt lËp tr×nh, th«ng qua gi¶i thuËt vµ kÕt qu¶ ®¸nh gi¸ thuËt to¸n mµ kh«ng chøng minh l¹i nh÷ng kÕt qu¶ ®ã, v× nã ®· ®îc tr×nh bµy trong mét chuy²n ®Ò kh¸c cña tin häc.

Nh÷ng thuËt to¸n s¾p xÕp vµ t×m kiÕm sÏ ®îc bµn luËn trong ch¬ng nµy bao gåm c¸c thuËt to¸n s¾p xÕp ®¬n gi¶n nh : chän trùc tiÕp (Selection), thuËt to¸n sñi bät (Bubble), thuËt to¸n chÌn trùc tiÕp (Insertion), c¸c thuËt to¸n s¾p xÕp nhanh nh quick sort, merge sort, heap sort. Trong

tÊt c¶ c¸c vÝ dô minh häa cho gi¶i thuËt s¾p xÕp vµ t×m kiÕm, chóng ta sÏ sö dông tËp c¸c sè nguyªn díi ®©y lµm vÝ dô s¾p xÕp. D·y sè nguyªn nµy sÏ kh«ng ®îc nh¾c l¹i trong khi gi¶i thÝch mçi thuËt to¸n s¾p xÕp.

42 23 74 11 65 58 94 36 99 87

7.2- Gi¶i thuËt Selection Sort

Néi dung cña Selection Sort lµ lÇn lît chän phÇn tö nhá nhÊt trong d·y chØ sè k_1, k_2,. . ., k_n víi i = 0, 1, . .,n; $k_i < k_{i+1} < . . ., k_n$ vµ ®æi chç cho phÇn tö thø k_i. Nh vËy, sau j =n-1 lÇn chän, chóng ta sÏ sã d·y kho¸ ®îc s¾p xÕp theo thø tù t¨ng dÇn. §èi víi d·y sè trªn, chóng ta sÏ thùc hiÖn nh sau:

LÇn chän thø 0: T×m trong kho¶ng tõ 0 ®Õn n-1 b»ng c¸ch thùc hiÖn n-1 lÇn so s¸nh ®Ó x¸c ®Þnh phÇn tö min_0 vµ ®æi chç cho phÇn tö ë vÞ trÝ 0.

LÇn chän thø 1: T×m trong kho¶ng tõ 1 ®Õn n-1 b»ng c¸ch thùc hiÖn n-2 lÇn so s¸nh ®Ó x¸c ®Þnh phÇn tö min_1 vµ ®æi chç cho phÇn tö ë vÞ trÝ 1.

. . . . . .

LÇn chän thø i: T×m trong kho¶ng tõ i ®Õn n-1 b»ng c¸ch thùc hiÖn n- i lÇn so s¸nh ®Ó x¸c ®Þnh phÇn tö min_i vµ ®æi chç cho phÇn tö ë vÞ trÝ i.

LÇn chän thø n-2: T×m trong kho¶ng tõ n-2 ®Õn n-1 b»ng c¸ch thùc hiÖn 1 lÇn so s¸nh ®Ó x¸c ®Þnh phÇn tö min_{n-2} vµ ®æi chç cho phÇn tö ë vÞ trÝ n-2.

§é phøc t¹p tÝnh to¸n cña gi¶i thuËt Selection Sort lµ:

$C_{min}=C_{max}=C_{tb} = n (n-1)/2$

Qu¸ tr×nh s¾p xÕp d·y sè ®îc minh häa th«ng qua b¶ng sau:

i	ki	1	2	3	4	5	6	7	8	9
0	**42**	**11**	**11**	**11**	**11**	**11**	**11**	**11**	**11**	**11**
1	**23**	**23**	**23**	**23**	**23**	**23**	**23**	**23**	**23**	**23**

2	**74**	**74**	**74**	**36**	**36**	**36**	**36**	**36**	**36**	**36**
3	**11**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**42**	**42**
4	**65**	**65**	**65**	**65**	**65**	**58**	**58**	**58**	**58**	**58**
5	**58**	**58**	**58**	**58**	**58**	**65**	**65**	**65**	**65**	**65**
6	**94**	**94**	**94**	**94**	**94**	**94**	**74**	**74**	**74**	**74**
7	**36**	**36**	**36**	**74**	**74**	**74**	**94**	**87**	**87**	**87**
8	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**99**	**94**	**94**
9	**87**	**87**	**87**	**87**	**87**	**87**	**87**	**94**	**99**	**99**

Ch¬ng tr×nh ®îc cµi ®Æt nh sau:

```
        }
        delay(1000);
}#include    <stdio.h>
#include     <conio.h>
#include     <stdlib.h>
#include     <alloc.h>
#include     <dos.h>
void Select(int *, int);
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);

void Select(int *A, int n){
        register i,j,temp;
        for(i=0;i<n-1;i++){
```

```
                for (j=i+1;j<n;j++){
                        if(A[i]>A[j]){
                                temp=A[i];
                                A[i]=A[j];
                                A[j]=temp;
                        }
                }
                In(A,n);
        }
}
void In(int *A, int n){
        register int i;
        for(i=0;i<n;i++)
                printf("%5d",A[i]);
        delay(1000);
}
void main(void){
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Select(A,n);
        free(A);
}
```

7.3- Gi¶i thuËt Insertion Sort// chen nhanh

Gi¶i thuËt Insert Sort ®îc thùc hiÖn dùa trªn kinh nghiÖm cña nh÷ng ngêi ch¬i bµi. Khi trªn tay cã i-1 l¸ bµi ®· ®îc s¾p xÕp ®ang ë trªn tay, nay ta thªm l¸ bµi thø i th× l¸ bµi ®ã ®îc so s¸nh víi l¸ bµi i-1, i-2, . . ®Ó t×m ®îc vÞ trÝ thÝch hîp vµ chÌn vµo qu©n bµi thø i.

Víi nguyªn t¾c s¾p bµi nh vËy, gi¶i thuËt ®îc thùc hiÖn nh sau:

LÊy phÇn tö ®Çu tiªn i_0, ®¬ng nhiªn tËp mét phÇn tö lµ tËp ®· ®îc s¾p xÕp.

LÊy tiÕp phÇn tö thø i_1 chän vÞ trÝ thÝch hîp cña phÇn tö thø i_1 trong tËp hai phÇn tö vµ thùc hiÖn ®æi chç.

. . . . . .

.

LÊy tiÕp phÇn tö thø i_k chän vÞ trÝ thÝch hîp cña phÇn tö thø i_k trong tËp hai i_{k-1} phÇn tö vµ thùc hiÖn ®æi chç, d·y sÏ ®îc s¾p xÕp hoµn toµn sau n-1 lÇn chÌn phÇn tö vµo vÞ trÝ thÝch hîp.

§é phøc t¹p bÐ nhÊt cña thuËt to¸n lµ: Cmin = (n-1);

§é phøc t¹p lín nhÊt cña thuËt to¸n lµ: $n(n-1)/2 = O(n^2)$

§é phøc t¹p trung b×nh cña thuËt to¸n lµ: $(n^2 +n- 2)/4 = O(n^2)$

Qu¸ tr×nh s¾p xÕp theo Insertion Sort ®îc m« t¶ nh sau:

Lît	**1**	**2**	**3**	**4**	**. . .**	**8**	**9**	**10**
Kho¸	**42**	**23**	**74**	**11**	**. . .**	**36**	**99**	**87**
1	**42**	**23**	**23**	**11**	**. . .**	**11**	**11**	**11**
2		**42**	**42**	**23**	**. . .**	**23**	**23**	**23**
3			**74**	**42**	**. . .**	**42**	**36**	**36**
4				**74**	**. . .**	**58**	**42**	**42**
5					**. . .**	**65**	**58**	**58**
6					**. . .**	**74**	**65**	**65**
7					**. . .**	**94**	**74**	**74**
8					**. . .**		**94**	**87**
9					**. . .**		**99**	**95**
10					**. . .**			**99**

ThuËt to¸n ®îc cµi ®Æt nh sau:

```
#include      <stdio.h>
#include      <conio.h>
#include      <stdlib.h>
#include      <alloc.h>
#include      <dos.h>
void Insert(int *, int);
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Insert(int *A, int n){
        register i,j,temp;
        for (i=1;i<n;i++){
                temp=A[i];
                for(j=i-1;j>=0 && temp<A[j];j--)// xet cho cac phan tu o phia
truoc i

                        A[j+1]=A[j];
                A[j+1]=temp;
                printf("\n");
                In(A,i+1);
        }
}
void In(int *A, int n){
```

```
        register int i;
        for(i=0;i<n;i++)
                printf("%5d",A[i]);
        delay(1000);
}
void main(void){
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Insert(A,n);
        free(A);
}
```

7.4- Gi¶i thuËt Bubble Sort

Gi¶i thuËt Bubble Sort ®îc thùc hiÖn b»ng c¸ch ®æi chç liªn tiÕp hai phÇn tö kÕ cËn khi chóng ngîc thø tù. Qu¸ tr×nh thùc hiÖn ®îc duyÖt tõ ®¸y lªn ®Ønh. Nh vËy, sau lÇn duyÖt thø nhÊt, phÇn tö lín nhÊt sÏ ®îc xÕp ®óng ë vÞ trÝ thø n-1, ë lÇn duyÖt thø k th× k phÇn tö lín nhÊt ®· ®îc xÕp ®óng vÞ trÝ n-1, n-2, . ., n-k+1. Sau lÇn duyÖt thø n-1, toµn bé n phÇn tö sÏ ®îc s¾p xÕp. Víi ph¬ng ph¸p nµy, c¸c phÇn tö cã gi¸ trÞ nhá ®îc næi dÇn lªn nh níc sñi bät nhê ®ã nã cã tªn gäi “ph¬ng ph¸p sñi bät”.

§é phøc t¹p cña thuËt to¸n Bubble Sort lµ:

$C_{min} = C_{max} = C_{tb} = n(n-1)/2.$

Ch¬ng tr×nh m« t¶ thuËt to¸n Bubble Sort ®îc cµi ®Æt nh sau:

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
void Bubble(int *, int);
void Init(int *, int);
```

```
void In(int *, int);
void Init(int *A, int n){
	int i;
	printf("\n Tao lap day so:");
	for (i=0; i<n;i++){
		A[i]=random(1000);
		printf("%5d",A[i]);
	}
	delay(1000);
}
void Bubble(int *A, int n){
	register i,j,temp;
	for (i=1; i<n; i++){
		for (j=n-1; j>=i; j--){
			if (A[j-1]>A[j]){
				temp=A[j-1];
				A[j-1]=A[j];
				A[j]=temp;
			}
		}
		printf("\n Ket qua lan:%d", i);
		In(A,n);
	}
}
void In(int *A, int n){
	register int i;
	for(i=0;i<n;i++)
		printf("%5d",A[i]);
	delay(1000);
}
void main(void){
```

```
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Bubble(A,n);
        free(A);
}
```

7.5- Gi¶i thuËt Shaker Sort

ThuËt to¸n Shaker Sort lµ c¶i tiÕn cña thuËt to¸n Bubble Sort. Trong ®ã, sau mçi lÇn duyÖt ®i ®Ó xÕp ®óng vÞ trÝ phÇn tö lín nhÊt, chóng ta thùc hiÖn duyÖt l¹i ®Ó s¾p ®óng vÞ trÝ phÇn tö nhá nhÊt. D·y sÏ ®îc s¾p sau [n/2] + 1 lÇn duyÖt. Ch¬ng tr×nh m« t¶ thuËt to¸n Shaker Sort ®îc thùc hiÖn nh sau:

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
void Shaker(int *, int);
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Shaker(int *A, int n){
        register i,j,temp, exchange;
```

```
	do {
		exchange=0;
		for (i=n-1; i>0; i--){// thuc hien viec sap xep theo phuong phap sui bot
			if (A[i-1]>A[i]){
				temp=A[i-1];
				A[i-1]=A[i];
				A[i]=temp;
				exchange=1;
			}
		}
		for(j=1; j<n;j++){// sau khi tim duoc phan tu lon nhat dat o vi tri cuoi thi tim phan tu nho nhat dat o vi tri dau tien
			if (A[j-1]>A[j]){
				temp=A[j-1];
				A[j-1]=A[j];
				A[j]=temp;
				exchange=1;
			}
		}
		printf("\n Ket qua lan:");
		In(A,n);
	}while(exchange);
}
void In(int *A, int n){
	register int i;
	for(i=0;i<n;i++)
		printf("%5d",A[i]);
	delay(1000);
}
void main(void){
```

```
    int *A,n;clrscr();
    printf("\n Nhap n="); scanf("%d",&n);
    A=(int *) malloc(n*sizeof(int));
    Init(A,n);Shaker(A,n);
    free(A);
}
```

7.6- Gi¶i thuËt Quick Sort

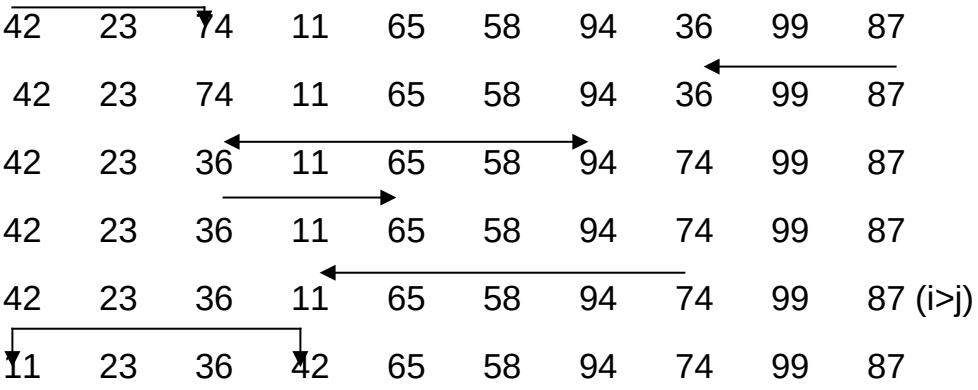
Ph¬ng ph¸p s¾p xÕp kiÓu ph©n ®o¹n lµ mét c¶i tiÕn cña ph¬ng ph¸p Selection Sort. §©y lµ mét ph¬ng ph¸p tèt do C.A.R. Hoare ®a ra vµ ®Æt tªn cho nã lµ gi¶i thuËt Quick Sort.

Néi dung chñ ®¹o cña ph¬ng ph¸p nµy lµ chän ngÉu nhiªn mét phÇn tö nµo ®ã cña d·y lµm khoÏ chèt. TÝnh tõ khoÏ chèt, cÏc phÇn tö nhá h¬n khoÏ ph¶i ®îc xÕp vµo trÝc chèt (®Çu d·y), mäi phÇn tö sau chèt ®îc xÕp vµo sau chèt (cuèi d·y). §Ó lµm ®îc viÖc ®ã, cÏc phÇn tö trong d·y sÏ ®îc so sÏnh víi khoÏ chèt vµ trÏo ®æi vÞ trÝ cho nhau, hoÆc cho khoÏ chèt nÕu phÇn tö ®ã lín h¬n chèt mµ l¹i n»m trÝc chèt hoÆc nhá h¬n chèt nhng l¹i n»m sau chèt. Khi viÖc ®æi chç lÇn ®Çu tiªn ®· thùc hiÖn xong th× d·y h×nh thµnh hai ®o¹n: mét ®o¹n bao gåm cÏc phÇn tö nhá h¬n chèt, mét ®o¹n gåm cÏc phÇn tö lín h¬n chèt, cßn chèt chÝnh lµ vÞ trÝ cña phÇn tö trong d·y ®îc s¾p xÕp.

Ïp dông kü thuËt nh trªn cho mçi ®o¹n trÝc chèt vµ sau chèt cho tíi khi cÏc ®o¹n cßn l¹i hai phÇn tö th× viÖc ghi nhí kh«ng cßn cÇn thiÕt n÷a. D·y sÏ ®îc s¾p xÕp khi tÊt c¶ cÏc ®o¹n ®îc xö lý xong. VÝ dô víi d·y :

42 23 74 11 65 58 94 36 99 87

Ta chän chèt ®Çu tiªn lµ 42. §Ó phÏt hiÖn ra hai khoÏ cÇn ®æi chç cho nhau, ta dïng hai biÕn i, j víi giÏ trÞ ban ®Çu i=2, j=10. NÕu k_i < 42 th× tiÕp tôc tï¨ng i vµ lÆp l¹i cho tíi khi gÆp phÇn tö thø k_i >42. DuyÖt cÏc phÇn tö thø k_j víi 42 nÕu k_j > 42 th× j gi¶m ®i mét, cho tíi khi gÆp phÇn tö thø k_j <42 th× phÇn tö thø k_i vµ k_j ®îc ®æi chç cho nhau. QuÏ tr×nh sÏ ®îc lÆp l¹i víi k_i vµ k_j cho tíi khi i=j chÝnh lµ vÞ trÝ dµnh cho khoÏ 42. Cuèi cïng chóng ta ®æi chç 42 cho khoÏ cho k_j.



Nh vËy, kÕt thóc lÇn thø nhÊt, chóng ta ®îc hai ®o¹n ®îc ph©n biÖt bëi kho¸ 42 nh sau:

(11 23 36) [42] (65 58 94 74 99 87)

Qu¸ tr×nh ®îc lÆp l¹i t¬ng tù cho tõng ph©n ®o¹n cho tíi khi d·y ®îc s¾p xÕp hoµn toµn. Chóng ta cã thÓ cµi ®Æt gi¶i thuËt b»ng viÖc sö dông stack hoÆc ®Ö qui.

§é phøc t¹p tÝnh to¸n cña gi¶i thuËt Quick Sort:

Trêng hîp tèt nhÊt $C_{max} = C_{tb} = O(n \log_2 n)$

Truêng hîp xÊu nhÊt $C_{min} = k.O(n^2)$

Sau ®©y lµ ch¬ng tr×nh cµi ®Æt gi¶i thuËt Quick Sort b»ng ph¬ng ph¸p ®Ö qui.

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
void qs(int *, int ,int);
void Quick(int *,int );
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
    int i;
```

```
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Quick(int *A, int n){
        qs(A,0,n-1);
}
void qs(int *A, int left,int right) {
        register i,j;int x,y;
        i=left; j=right;
        x= A[(left+right)/2];
        do {
                while(A[i]<x && i<right) i++;
                while(A[j]>x && j>left) j--;
                if(i<=j){
                        y=A[i];A[i]=A[j];A[j]=y;
                        i++;j--;
                }
        } while (i<=j);
        if (left<j) qs(A,left,j);
        if (i<right) qs(A,i,right);
}
void In(int *A, int n){
        register int i;
        for(i=0;i<n;i++)
                printf("%5d",A[i]);
        delay(1000);
}
```

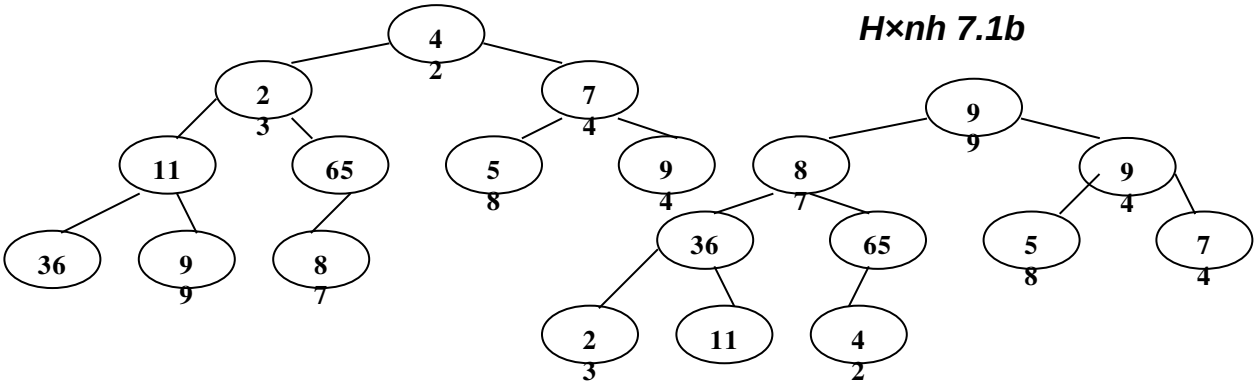
```
void main(void){
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *)malloc(n*sizeof(int));
        Init(A,n);Quick(A,n);printf("\n");
        In(A,n);getch();
        free(A);
}
```

7.7- Gi¶i thuËt Heap Sort

Heap lµ mét c©y nhÞ ph©n ®îc biÓu diÔn b»ng mét m¶ng, m¶ng ®ã biÓu diÔn mét c©y nhÞ ph©n hoµn chØnh sao cho khãa ë node cha bao giê còng lín h¬n kho¸ cña node con cña nã.

S¾p xÕp kiÓu Heap Sort ®îc tiÕn hµnh qua hai giai ®o¹n. Giai ®o¹n ®Çu tiªn c©y nhÞ ph©n biÓu diÔn b¶ng kho¸ ®îc biÕn ®æi ®Ó ®a vÒ mét heap. Nh vËy, ®èi víi heap, nÕu j lµ chØ sè cña node con th× [j/2] lµ chØ sè cña node cha. Theo ®Þnh nghÜa cña heap th× node con bao giê còng nhá h¬n node cha. Nh vËy, node gèc cña heap lµ khãa cã gi¸ trÞ lín nhÊt trong mäi node. VÝ dô c©y ban ®Çu lµ c©y 7.1a th× heap cña nã lµ 7.1b.

H×nh 7.1a



H×nh 7.1b

§Ó chuyÓn c©y nhÞ ph©n 7.1a thµnh c©y nhÞ ph©n 7.1b lµ mét heap, chóng ta thùc hiÖn duyÖt tõ díi l[a]n (bottom up). Node l¸ ®¬ng nhi[a]n lµ mét heap. NÕu c©y con b[a]n tr¸i vµ c©y con b[a]n ph¶i ®Òu lµ mét heap th× toµn bé c©y còng lµ mét heap. Nh vËy, ®Ó t[1]o thµnh heap, chóng ta thùc hiÖn so s¸nh néi dung node b[a]n tr¸i, néi dung node b[a]n ph¶i víi node cha cña nã, node nµo cã gi¸ trÞ lín h¬n sÏ ®îc thay ®æi lµm néi dung cña node cha. Qu¸ tr×nh lÇn ngîc l[1]i cho tíi khi gÆp node gèc, khi ®ã néi dung node gèc chÝnh lµ kho¸ cã gi¸ trÞ lín nhÊt.

Giai ®o[1]n thø hai cña gi¶i thuËt lµ ®a néi dung cña node gèc vÒ vÞ trÝ cuèi cïng vµ néi dung cña node cuèi cïng ®îc thay vµo vÞ trÝ node gèc, sau ®ã coi nh node cuèi cïng nh ®· bÞ lo[1]i bá v× thùc tÕ node cuèi cïng lµ gi¸ trÞ lín nhÊt trong d·y sè.

C©y míi ®îc t[1]o ra (kh«ng kÓ phÇn tö lo[1]i bá) kh«ng ph¶i lµ mét heap, chóng ta l[1]i thùc hiÖn vun thµnh ®èng vµ thùc hiÖn t¬ng tù nh tr[a]n cho tíi khi ®èng cßn mét phÇn tö lµ phÇn tö bÐ nhÊt cña d·y.

§é phøc t[1]p thuËt to¸n cña Heap Sort

$C_{max} = C_{tb} = O (n \log_2 n)$

Gi¶i thuËt Heap Sort ®îc cµi ®Æt nh sau:

```
#include    <stdio.h>
#include    <conio.h>
```

```
#include      <stdlib.h>
#include      <alloc.h>
#include      <dos.h>
void Heap(int *, int );
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Heap(int *A, int n) {
        int k,x,s,f,ivalue;
        for(k=1;k<n;k++){
                x=A[k];
                s=k; f=(s-1)/2;
                while(s>0 && A[f]<x){
                        A[s]=A[f];
                        s=f; f=(s-1)/2;
                }
                A[s]=x;
        }
        for(k=n-1;k>0;k--){
                ivalue=A[k];
                A[k]=A[0];
                f=0;
                if(k==1)
```

```
                s=-1;
            else
                s=1;
            if(k>2 && A[2]>A[1])
                s=2;
            while(s>=0 && ivalue<A[s]){
                A[f]=A[s];
                f=s;s=2*f +1;
                if (s+1<=k-1 && A[s]<A[s+1])
                    s=s+1;
                if (s>k-1)
                    s=-1;
            }
            A[f]=ivalue;
        }
}

void In(int *A, int n){
        register int i;
        for(i=0;i<n;i++)
            printf("%5d",A[i]);
        delay(1000);
}
void main(void){
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Heap(A,n);printf("\n");
        In(A,n);getch();
        free(A);
}
```

7.8- Gi¶i thuËt Merge Sort

S¾p xÕp theo Merge Sort lµ ph¬ng ph¸p s¾p xÕp b»ng c¸ch trén hai danh s¸ch ®· ®îc s¾p xÕp thµnh mét danh s¸ch ®· ®îc s¾p xÕp. Ph¬ng ph¸p Merge Sort ®îc tiÕn hµnh th«ng qua c¸c bíc nh sau:

Bíc 1: Coi danh s¸ch lµ n danh s¸ch con mçi danh s¸ch con gåm mét phÇn tö, nh vËy c¸c danh s¸ch con ®· ®îc s¾p xÕp. Trén tõng cÆp hai danh s¸ch con kÕ cËn thµnh mét danh s¸ch cã hai phÇn tö ®· ®îc s¾p xÕp, chóng ta nhËn ®îc n/2 danh s¸ch con ®· ®îc s¾p xÕp.

Bíc 2: Xem danh s¸ch cÇn s¾p xÕp nh n/2 danh s¸ch con ®· ®îc s¾p xÕp. Trén cÆp hai danh s¸ch kÕ cËn thµnh tõng danh s¸ch cã 4 phÇn tö ®· ®îc s¾p xÕp, chóng ta nhËn ®îc n/4 danh s¸ch con.

. . . . . .

Bíc thø i: Lµm t¬ng tù nh bíc i- 1. Qu¸ tr×nh ®îc tiÕp tôc khi chóng ta nhËn ®îc danh s¸ch cã n phÇn tö ®· ®îc s¾p xÕp. VÝ dô víi d·y:

42 23 74 11 68 58 94 36

lÇn 1: [23 42] [11 74] [58 68] [94 36]

lÇn 2: [11 23 42 74] [36 58 68 94]

lÇn 3: [11 23 42 36 58 68 74 94]

Ch¬ng tr×nh cµi ®Æt gi¶i thuËt Merge Sort ®îc thùc hiÖn nh sau:

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
#define         MAX  10
```

```
void Merge(int *, int );
void Init(int *, int);
void In(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Merge(int *A, int n) {
        int i,j,k,low1,up1,low2,up2,size;
        int *dstam;size=1;dstam=(int *) malloc(n*sizeof(int));
        while(size<n){
                low1=0;k=0;
                while(low1 +size <n){
                        low2=low1+size;    up1=low2-1;
                        if (low2+size-1< n)
                                up2=low2+size-1;
                        else
                                up2=n-1;
                        for(i=low1, j=low2; i<=up1 && j<=up2; k++){
                                if(A[i]<=A[j])
                                        dstam[k]=A[i++];
                                else
                                        dstam[k] =A[j++];
                        }
                        for(;i<=up1;k++)
                                dstam[k]=A[i++];
```

```
                    for(;j<=up2;k++)
                        dstam[k]=A[j++];
                    low1=up2+1;
                }
                for (i=low1; k<n;i++)
                    dstam[k++]=A[i];
                for(i=0;i<n;i++)
                    A[i]=dstam[i];
                size*=2;
        }
        printf("\n Ket qua:");
        In(A,n);free(dstam);
}
void In(int *A, int n){
        register int i;
        for(i=0;i<n;i++)
                printf("%5d",A[i]);
        delay(1000);
}
void main(void){
        int *A,n;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Merge(A,n);printf("\n");
        free(A);
}
```

7.9- T×m kiÕm (Searching)

T×m kiÕm lµ c«ng viÖc quan träng ®èi víi c¸c hÖ thèng tin häc vµ cã liªn quan mËt thiÕt víi qu¸ tr×nh s¾p xÕp d÷ liÖu. Bµi to¸n t×m kiÕm tæng qu¸t cã thÓ ®îc ph¸t biÓu nh sau:

" Cho mét b¶ng gåm n b¶n ghi R_1, R_2, . ., R_n. Víi mçi b¶n ghi R_i ®îc t-¬ng øng víi mét kho¸ k_i (trêng thø i trong record). H·y t×m b¶n ghi cã gi¸ trÞ cña kho¸ b»ng X cho tríc".

NÕu chóng ta t×m ®îc b¶n ghi cã gi¸ trÞ khãa lµ X th× phÐp t×m kiÕm ®îc tho¶ (successful). NÕu kh«ng cã gi¸ trÞ khãa nµo lµ X, th× qu¸ tr×nh t×m kiÕm lµ kh«ng tho¶ (unsuccessful). Sau qu¸ tr×nh t×m kiÕm, cã thÓ xuÊt hiÖn yªu cÇu bæ xung thªm b¶n ghi míi cã gi¸ trÞ khãa lµ X th× gi¶i thuËt ®îc gäi lµ gi¶i thuËt t×m kiÕm bæ sung.

7.9.1- T×m kiÕm tuÇn tù (Sequential Searching)

T×m kiÕm tuÇn tù lµ kü thuËt t×m kiÕm cæ ®iÓn trªn mét danh s¸ch cha ®îc s¾p xÕp. Néi dung c¬ b¶n cña ph¬ng ph¸p t×m kiÕm tuÇn tù lµ duyÖt tõ b¶n ghi thø nhÊt cho tíi b¶n ghi cuèi cïng, vµ so s¸nh lÇn lît gi¸ trÞ cña kho¸ víi gi¸ trÞ X cÇn t×m. Trong qu¸ tr×nh duyÖt, nÕu cã b¶n ghi trïng víi gi¸ trÞ X th× chóng ta ®a ra vÞ trÝ cña b¶n ghi trong d·y, nÕu duyÖt tíi cuèi d·y mµ kh«ng cã b¶n ghi nµo cã gi¸ trÞ cña kho¸ trïng víi X th× qu¸ tr×nh t×m kiÕm tr¶ l¹i gi¸ trÞ -1 (-1 ®îc hiÓu lµ gi¸ trÞ kho¸ X kh«ng thuéc d·y). Ch¬ng tr×nh cµi ®Æt ph¬ng ph¸p t×m kiÕm tuÇn tù ®îc thùc hiÖn nh sau:

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
int Sequential(int *, int, int);
void Init(int *, int);
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
```

```
        }
        delay(1000);
}
int Bubble(int *A, int x, int n){
        register i,temp;
        for (i=0; i<n ; i ++){
                if (A[i] == X)
                        return(i);
        }
        return(-1);
}
void main(void){
        int *A,n, x, k;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        printf("\n Sè x cÇn t×m:"); scanf("%d", &x);
        A=(int *) malloc(n*sizeof(int));
        k= Sequential(A,x,n);
        if ( k>=0)
                printf("\n %d ë vÞ trÝ %d", x,k);
        else
                printf("\n %d kh«ng thuéc d·y");
        free(A); getch();
}
```

7.9.2- T×m kiÕm nhÞ ph©n (Binary Searching)

T×m kiÕm nhÞ ph©n lµ ph¬ng ph¸p t×m kiÕm phæ biÕn ®îc thùc hiÖn trªn mét d·y ®· ®îc s¾p thø tù. Néi dung cña gi¶i thuËt ®îc thùc hiÖn nh sau: lÊy khãa cÇn t×m kiÕm X so s¸nh víi néi dung cña khãa cña phÇn tö ë gi÷a, vÞ trÝ cña phÇn tö ë gi÷a lµ mid = (low + hight)/ 2, trong ®ã cËn díi low =0, cËn trªn hight = n-1. V× d·y ®· ®îc s¾p xÕp nªn nÕu néi dung cña khãa t¹i vÞ trÝ gi÷a lín h¬n X th× phÇn tö cÇn t×m thuéc kho¶ng [mid+1, hight], nÕu néi dung cña khãa t¹i vÞ trÝ gi÷a nhá h¬n X th× phÇn tö

cÇn t×m thuéc kho¶ng [low, mid-1], nÕu néi dung cña khãa t¹i vÞ trÝ gi÷a trïng víi X th× ®ã chÝnh lµ phÇn tö cÇn t×m. ë bíc tiÕp theo, nÕu néi dung cña khãa t¹i vÞ trÝ gi÷a lín h¬n X th× ta dÞch chuyÓn cËn díi low lªn vÞ trÝ mid+ 1, nÕu néi dung cña khãa t¹i vÞ trÝ gi÷a nhá h¬n X th× ta dÞch chuyÓn cËn trªn vÒ vÞ trÝ mid- 1. Qu¸ tr×nh ®îc lÆp l¹i cho tíi khi gÆp khãa cã néi dung trïng víi X hoÆc cËn díi vît qu¸ cËn trªn hay X kh«ng thuéc d·y. ThuËt to¸n t×m kiÕm nhÞ ph©n ®îc minh häa nh sau:

```
int     Binary_Search( int *A, int X, int n){
        int mid, low=0, hight = n-1;
        while ( low<=hight ){ // lÆp nÕu cËn díi vÉn nhá h¬n cËn trªn
                mid = (low + hight) /2; // x¸c ®Þnh vÞ trÝ phÇn tö ë gi÷a
                if  (X > A[mid] )      low = mid +1; // X thuéc [mid+1, hight]
                else if (X < A[mid] ) hight = mid- 1; // X thuéc [low, mid-1]
                else    return(mid);
        }
        return(-1); // X kh«ng thuéc d·y
}
```

Ch¬ng tr×nh cô thÓ ®îc cµi ®Æt nh sau:

```
#include    <stdio.h>
#include    <conio.h>
#include    <stdlib.h>
#include    <alloc.h>
#include    <dos.h>
int     Binary_Search( int *, int, int);
void Bubble(int *, int);
void Init(int *, int);

int     Binary_Search( int *A, int X, int n) {
        int mid, low = 0, hight = n-1;
```

```
        while (low<=hight){
                mid = (low +hight)/2;
                if (X >A[mid] )       low = mid +1;
                else if (X<A[mid] ) hight = mid -1;
                else   return (mid);
        }
        return(-1);
}
void Init(int *A, int n){
        int i;
        printf("\n Tao lap day so:");
        for (i=0; i<n;i++){
                A[i]=random(1000);
                printf("%5d",A[i]);
        }
        delay(1000);
}
void Bubble(int *A, int n){
        register i,j,temp;
        for (i=1; i<n; i++){
                for (j=n-1; j>=i;j--){
                        if (A[j-1]>A[j]){
                                temp=A[j-1];
                                A[j-1]=A[j];
                                A[j]=temp;
                        }
                }
                printf("\n Ket qua lan:%d", i);
                In(A,n);
        }
}
```

```
void In(int *A, int n){
        register int i;
        for(i=0;i<n;i++)
                printf("%5d",A[i]);
        delay(1000);
}
void main(void){
        int *A,n, X, k;clrscr();
        printf("\n Nhap n="); scanf("%d",&n);
        printf("\n Sè cÇn t×m X="); scanf("%d",&X);
        A=(int *) malloc(n*sizeof(int));
        Init(A,n);Bubble(A,n); k= Binary_Search(A, X, n);
        if ( k>0)
                printf ("\n %d ë vÞ trÝ sè %d", X, k);
        else
                printf("\n %d kh«ng thuéc d·y");
        getch();
        free(A);
}
```

Bµi tËp ch¬ng 7

7.1. Cµi ®Æt ch¬ng tr×nh theo thuËt to¸n Quick Sort kh«ng dïng ph¬ng ph¸p ®Ö qui mµ dïng cÊu tróc stack.

7.2. T×m hiÓu vÒ gi¶i thuËt Shell-Sort lµ ph¬ng ph¸p c¶i tiÕn cña Insertion Sort.

7.3. Cµi ®Æt l¹i gi¶i thuËt Bubble Sort sao cho c¸c node nhá ®îc ®Èy dÇn vÒ phÝa tríc.

7.4. Mét Ternary Heap lµ c©y tam ph©n gÇn ®Çy ®îc cµi ®Æt b»ng m¶ng mét chiÒu, mçi node cã ba node con. Néi dung cña node cha bao giê còng lín h¬n hoÆc b»ng néi dung cña node con, c¸c node ®îc ®¸nh sè tõ 0 ®Õn n-1, node i cã 3 con lµ 3i+1, 3i+2, 3i+3. H·y cµi ®Æt gi¶i thuËt Ternary Heap.

7.5. Cµi ®Æt gi¶i thuËt Bubble Sort trªn file.

7.6. Cµi ®Æt gi¶i thuËt Insertion Sort trªn file.

7.7. Cµi ®Æt gi¶i thuËt Quick Sort trªn file.

7.8. Cµi ®Æt c¸c gi¶i thuËt s¾p xÕp theo nhiÒu kho¸ kh¸c nhau.

7.9. Nghiªn cøu vµ cµi ®Æt thuËt to¸n t×m kiÕm tam ph©n.

7.10. Nghiªn cøu vµ cµi ®Æt thuËt to¸n s¾p xÕp kiÓu hoµ nhËp thùc hiÖn trªn file.

7.11. ViÕt ch¬ng tr×nh chuyÓn ®æi mét file d÷ liÖu ®îc tæ chøc theo khu«n d¹ng *.DBF thµnh file kiÓu text. Ngîc l¹i, chuyÓn ®æi file d÷ liÖu kiÓu text thµnh mét file d÷ liÖu theo khu«n d¹ng DBF.

7.12. T×m hiÓu c¸ch s¾p xÕp vµ t×m kiÕm theo kiÓu index cña c¸c hÖ qu¶n trÞ c¬ së d÷ liÖu nh foxprol hoÆc access.