**Bài tập chương 4**

**Câu 1.(3 điểm)**

Cho dãy số nguyên X = { 8, 21, 7, 56, 10, -35, 19}

*Yêu cầu:* - Minh họa việc sắp xếp dãy X theo chiều tăng dần bằng phương pháp nổi bọt.

Qua mỗi một lần duyệt, ta được 1 phần tử đứng đúng vị trí

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần duyệt | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 8 | 21 | 7 | 56 | 10 | -35 | 19 |
| i=0 | 8  8  8  8  8  **-35** | 21  21  21  21  **-35**  **8** | 7  7  7  **-35**  **21**  21 | 56  56  **-35**  **7**  7  7 | 10  -**35**  **56**  56  56  56 | -**35**  **10**  10  10  10  10 | **19**  19  19  19  19  19 | So sánh cặp x5, x6 và giữ nguyên  So sánh cặp x4,x5 và đổi chỗ  So sánh cặp x3, x4 và đổi chỗ  So sánh cặp x1, x2 và đổi chỗ  So sánh cặp x1, x2 và đổi chỗ  So sánh cặp x0, x1 và đổi chỗ |
| i = 1 | -35 | **7** | 8 | 21 | 10 | 56 | 19 | Tương tự với lần duyệt i=0 |
| i = 2 | -35 | 7 | **8** | 10 | 21 | 19 | 56 |  |
| i = 3 | -35 | 7 | 8 | **10** | 19 | 21 | 56 |  |
| i = 4 | -35 | 7 | 8 | 10 | **19** | 21 | 56 |  |
| i = 5 | -35 | 7 | 8 | 10 | 19 | **21** | 56 | Dãy được sắp xếp |

-Viết giải thuật sắp xếp dãy X theo chiều tăng dần bằng phương pháp nổi bọt.

Void BubleSort (int X [ ], int n)

{ For (int i = 0 ; i < n – 1 ; i++)

For (int j = n – 1 ; j > i ; j--)

If (X [ j ] > X [ j – 1 ]  )

{

int tg = X [ j ] ;

X [ j ] = X [ j – 1 ] ;

X [ j – 1 ] = tg ;

}

}

**Câu 2.(3 điểm)** Cho dãy số nguyên X = {15, 20, 7, 56, 10, -35, 19}

*Yêu cầu:*- Minh họa việc sắp xếp dãy X theo chiều tăng dần bằng phương pháp lựa chọn.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần duyệt | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 15 | 20 | 7 | 56 | 10 | **-35** | 19 |
| i = 0 | -35 | 20 | **7** | 56 | 10 | 15 | 19 | Duyệt từ X0 đến X6, X5 nhỏ nhất đổi chỗ cho X0 |
| i = 1 | -35 | 7 | 20 | 56 | **10** | 15 | 19 | Duyệt từ X1 đến X6, X2 nhỏ nhất đổi chỗ cho X1 |
| i = 2 | -35 | 7 | 10 | 56 | 20 | **15** | 19 | Duyệt từ X2 đến X6, X4 nhỏ nhất đổi chỗ cho X2 |
| i = 3 | -35 | 7 | 10 | 15 | 20 | 56 | **19** | Duyệt từ X3 đến X6, X5 nhỏ nhất đổi chỗ cho X3 |
| i = 4 | -35 | 7 | 10 | 15 | 19 | 56 | **20** | Duyệt từ X4 đến x6, X6 nhỏ nhất đổi chỗ cho X5 |
| i = 5 | -35 | 7 | 10 | 15 | 19 | 20 | 56 | Dãy được sắp xếp |

-Viết giải thuật sắp xếp dãy X theo chiều tăng dần bằng phương pháp lựa chọn.

Void SelectSort ( SV ) a [ ] , int n )

{

SV tg ;

For ( int i = 0 ; i < n – 1 ; j < n ; i++ )

{

int m = i ;

for ( int j = i + 1 ; j < n ; j++)

if ( a [ j ] . tdiem < a [ m ] . tdiem )

m = j ;

if ( m ! = i )

{

tg = a [ i ] ;

a [ i ] = a [m ] ;

a [ m ] = tg ;

}

}

}

**Câu 3.(3 điểm)**

Cho dãy số nguyên X = {15, 20, 7, 56, 10, -35, 19}

*Yêu cầu:*

- Minh họa việc sắp xếp dãy X theo chiều tăng dần bằng phương pháp chèn.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần chia | X0 | X1 | X2 | X3 | X4 | X5 | X6 | Giải thích |
| 15 | 20 | 7 | 56 | 10 | -35 | 19 |
| i = 0 | 15 | **20** | 7 | 56 | 10 | -35 | 19 | Lấy t=X1=20, so sánh với X0. Cả 2 đều đã trong thứ tự tăng dần |
| i = 1 | 15 | 20 | **7** | 56 | 10 | -35 | 19 | Lấy t=X2=7 chèn vào dãy đích |
|  | **15** | 15 | 20 | 56 | 10 | -35 | 19 |  |
| i = 2 | 7 | 15 | 20 | **56** | 10 | -35 | 19 | Lấy t=X3=56 chèn vào dãy đích |
| i = 3 | 7 | 15 | 20 | 56 | **10** | -35 | 19 | Lấy t=X4=10 chèn vào dãy đích |
|  | 7 | **15** | 15 | 20 | 56 | -35 | 19 |  |
| i = 4 | 7 | 10 | 15 | 20 | 56 | **-35** | 19 | Lấy t=X5=-35 chèn vào dãy đích |
|  | **7** | 7 | 10 | 15 | 20 | 56 | 19 |  |
| i = 5 | -35 | 7 | 10 | 15 | 20 | 56 | **19** | Lấy t=X6=19 chèn vào dãy đích |
|  | -35 | 7 | 10 | 15 | **20** | 20 | 56 |  |
|  | -35 | 7 | 10 | 15 | 19 | 20 | 56 | Dãy được sắp xếp |

- Viết giải thuật sắp xếp dãy X theo chiều tăng dần bằng phương pháp chèn.

Void InsertSort (int X [ ] , int n)

{

For (int i =1 ; i < n ; i++)

{

int t = X [ i ] ; / / t chứa phần tử đang xét

int j i – 1 ;

/ / chuyển các phần tử để tìm vị trí chèn

While (j > = 0 && t < X [ j ] )

{

X [ j + 1] = X [ j ] ;

j = j -1 ;

}

/ / Đưa phân tử t vào vị trí

X [ j + 1 ] = t ;

}

}

**Bài tập chương 6**

**Câu 1.(2 điểm)**

Cho dãy khóa X = {9, 21, 7, 60, 10, -35, 15, 18, 45}

Mô tả từng bước bằng hình vẽ quá trình tạo cây nhị phân tìm kiếm.

9

B1 : Chọn khóa 9 làm gốc.

Ta có : 21>9 => 21 nằm bên phải gốc 9

7<9 => 7 nằm bên trái gốc 9

21

7

B2 : với nút giá trị 60 ta có : 60>9 => nút 60 nằm bên phải gốc.

Mà nút phải của gốc là nút 21 (khác rỗng)

Tiếp tục so sánh : 60>21 =>60 nằm bên phải nút 21

9

7

21

60

B3 : Xét khóa 10 ta có : 10>9 => nút 10 duyệt về bên cây con phải của gốc 9

Mà nút phải của gốc 9 khác rỗng, tiếp tục so sánh: 10<21 => 10 thuộc cây con phải của nút 21

9

7

21

10

60

B4 : Xét khóa -35 ta có : -35<9 => nút -35 thuộc cây con trái của gốc 9.

Nút trái của gốc 9 khác rỗng, tiếp tục so sánh : -35<7 => -35 nằm về phía bên trái của nút 7

9

7

21

-35

10

60

B5 : Xét khóa 15 ta có : 15>9 => duyệt về phía bên phải của gốc 9

Nút phải của gốc 9 khác rỗng, so sánh: 15<21 => duyệt nút 15 về phía bên trái nút 21.

Nút trái của 21 khác rỗng, so sánh 15>10 => nút 15 nằm phía bên phải nút 21

9

7

21

-35

10

60

15

B6 : khóa có giá trị 18 :

Tương tự, ta so sánh : 18>9 => 18 nằm phía bên phải gốc 9.

Vì nút phải gốc 9 khác rỗng, tiếp tục so sánh :

18<21 => khóa 18 nằm bên phải nút 21 (nút phải của 21 khác rỗng)

* So sánh : 18>10 =>khóa 18 nằm bên phải nút 10 (nút phải 10 khác rỗng)
* So sánh :18>15 => khóa 18 nằm bên phải nút 15

9

7

21

-35

10

60

15

18

B7 : xét khóa cuối cùng : khóa 45

Ta có : 45>9 => khóa 45 nằm bên phải nút gốc.

* 45>21 => 45 nằm bên phải nút 21
* 45<60 => 45 nằm bên trái nút 60.

🡺Ta thu được kết quả cây nhị phân tìm kiếm :

9

21

7

-35

60

10

15

45

18

❖Mô tả thao tác hủy nút có khóa là 21 trong cây nhị phân tìm kiếm trên

Vì nút cần xóa (nút 21) là nút có cả cây con trái và cây con phải

* Tìm nút thay thế cho nút cần xóa
* C1 : chọn nút thay thế là nút trái nhất của cây con phải (nút 18)
* C2 : chọn nút thay thế là nút phải nhất của cây cây con trái (nút 60)

🟅chọn nút thay thế là nút 18 – nút phải nhất của cây con trái ( nút có giá trị lớn nhất trên cây con trái của nút cần xóa )

⮱ Kết quả sau khi loại bỏ nút 21 :

9

18

7

60

10

-35

45

15

**Câu 2.(2 điểm)**

Cho dãy khóa X = {15, 21, 7, 60, 2, 65, 1, 10, 45}

Mô tả từng bước bằng hình vẽ quá trình tạo cây nhị phân tìm kiếm.

15

B1 : Chọn khóa 15 làm gốc.

Ta có : 21>15 => 21 nằm bên phải gốc 15

7<15 => 7 nằm bên trái gốc 15

21

7

B2 : với nút giá trị 60 ta có : 60>15 => nút 60 nằm bên phải gốc.

Mà nút phải của gốc là nút 21 (khác rỗng)

Tiếp tục so sánh : 60>21 =>60 nằm bên phải nút 21

15

7

21

60

B3 : Xét khóa 2 ta có : 2<15 => nút 2 duyệt về bên cây con trái của gốc 15

Mà nút trái của gốc 15 khác rỗng, tiếp tục so sánh: 2<7 => 2 thuộc cây con trái của nút 7

15

7

21

2

60

B4 : Xét khóa 65 ta có : 65>15 => nút 65 thuộc cây con phải của gốc 15.

Nút phải của gốc 15 khác rỗng, tiếp tục so sánh : 65>21 => 65 nằm về phía bên phải của nút 21.

Nút phải của 21 khác rỗng, tiếp tục so sánh : 65>60 => 65 nằm về phía bên phải của nút 60

15

7

21

2

60

65

B5 : Xét khóa 1 ta có : 1<15=> duyệt về phía bên trái của gốc 15

Nút trái của gốc 15 khác rỗng, so sánh: 1<7 => duyệt nút 1 về phía bên trái nút 7.

Nút trái của 7 khác rỗng, so sánh 1<2 => nút 1 nằm phía bên trái nút 2

15

7

21

2

60

1

65

B6 : khóa có giá trị 10 :

Tương tự, ta so sánh : 10<15 => 10 nằm phía bên trái gốc 15.

Vì nút trái gốc 15 khác rỗng, tiếp tục so sánh: 10>7=> khóa 10 nằm bên phải nút 7

15

7

21

10

2

60

65

1

B7 : xét khóa cuối cùng : khóa 45

Ta có : 45>15 => khóa 45 nằm bên phải nút gốc. mà nút phải gốc 15 khác rỗng

* So sánh : 45>21 => 45 nằm bên phải nút 21. Mà nút phải nút 21 khác rỗng
* So sánh : 45<60 => 45 nằm bên trái nút 60.

🡺Ta thu được kết quả cây nhị phân tìm kiếm :

155

21

7

2

60

10

45

1

65

❖Mô tả thao tác hủy nút có khóa là 60 trong cây nhị phân tìm kiếm trên

Vì nút cần xóa ( nút 60 ) là nút có cả cây con trái và câu con phải

* Ta phải tìm nút thay thế cho nút cần xóa
* C1 : chọn nút trái nhất trên cây con phải ( nút 45)
* C2 : chọn nút phải nhất trên cây con trái (nút 65)

🟅chọn nút thay thế là nút 45 – nút phải nhất của cây con trái ( nút có giá trị lớn nhất trên cây con trái của nút cần xóa )

⮱ Ta thu được cây sau khi xóa nút 60 :

15

7

21

10

2

45

65

1

**Câu 3.(2 điểm)**

Cho dãy khóa X = {15, 21, 7, 60, 10, 65, 1, 9, 45}

Mô tả từng bước bằng hình vẽ quá trình tạo cây nhị phân tìm kiếm.

15

B1 : Chọn khóa 15 làm gốc.

Ta có : 21>15 => 21 nằm bên phải gốc 15

7<15 => 7 nằm bên trái gốc 15

21

7

B2 : với nút có giá trị 60 ta có : 60>15 => nút 60 nằm bên phải gốc.

Mà nút phải của gốc là nút 21 (khác rỗng)

Tiếp tục so sánh : 60>21 =>60 nằm bên phải nút 21

15

7

21

60

B3 : Xét khóa 10 ta có : 10<15 => nút 10 duyệt về bên cây con trái của gốc 15

Mà nút trái của gốc 15 khác rỗng, tiếp tục so sánh: 10>7 => 10 thuộc cây con phải của nút 7

15

7

21

10

60

B4 : Xét khóa 65 ta có : 65>15 => nút 65 thuộc cây con phải của gốc 15.

Nút phải của gốc 15 khác rỗng, tiếp tục so sánh : 65>21 => 65 nằm về phía bên phải của nút 21.

Nút phải của 21 khác rỗng, tiếp tục so sánh : 65>60 => 65 nằm về phía bên phải của nút 60

15

7

21

10

60

65

B5 : Xét khóa 1 ta có : 1<15=> duyệt về phía bên trái của gốc 15

Nút trái của gốc 15 khác rỗng, so sánh: 1<7 => duyệt nút 1 về phía bên trái nút 7.

15

7

21

1

10

60

65

B6 : khóa có giá trị 9:

Tương tự, ta so sánh : 9<15 => 9 nằm phía bên trái gốc 15.

Vì nút trái gốc 15 khác rỗng, tiếp tục so sánh: 9>7=> khóa 9 nằm bên phải nút 7

Vì nút 7 khác rỗng, tiếp tục so sánh : 9<10 => giá trị 9 nằm bên trái nút 10

15

7

21

10

1

60

9

65

B7 : xét khóa cuối cùng : khóa 45

Ta có : 45>15 => khóa 45 nằm bên phải nút gốc. mà nút phải của gốc 15 khác rỗng

* So sánh : 45>21 => 45 nằm bên phải nút 21. Mà nút phải nút 21 khác rỗng
* So sánh : 45<60 => 45 nằm bên trái nút 60.

🡺Ta thu được kết quả cây nhị phân tìm kiếm :

155

21

7

1

60

10

9

45

65

❖Mô tả thao tác hủy nút có khóa là 7 trong cây nhị phân tìm kiếm trên

Vì nút cần xóa (nút 7) là nút có cả cây con trái và cây con phải

=> Ta phải tìm nút thay thế cho nút cần xóa

* C1 : Chọn nút thay thế là nút có giá trị lớn nhất trên cây con trái (nút 1)
* C2 : chọn nút thay thế là nút có giá trị nhỏ nhất trên cây con phải (nút 9)

🟅Chọn nút thay thế là nút 9 – nút có giá trị nhỏ nhất trên cây con phải

⮱Ta thu được kết quả sau khi xóa nút 7 :

155

9

21

1

60

10

45

65

**Câu 4 (2,0 điểm):** Cho biểu thức số học S = (a + b!) \* c – a / (d + e). Hãy thực hiện các yêu cầu sau:

1. Dựng cây nhị phân biểu diễn biểu thức S.

-

/

\*

+

a

+

c

!

e

d

a

b

1. Viết lại biểu thức S dưới dạng tiền tố, hậu tố.

🟅Biểu thức tiền tố:

- \* + a b! c / a + d e

🟅Biểu thức hậu tố:

a b! + c \* a d e + / -

1. Mô tả quá trình định giá biểu thức bằng ngăn xếp.

Định giá biểu thức: a b! + c \* a d e + / - bằng ngăn xếp

b!=A a + A = B c \* B = C d + e = D a / D = E C – E = K

e

d D

b A B a a E

a a c C C C K

**Câu 5 (2,0 điểm):** Cho biểu thức số học S = (sinx + b) \* c \* d + a \* b. Hãy thực hiện các yêu cầu sau:

* 1. Dựng cây nhị phân biểu diễn biểu thức.

\*\*\*

\*

+

b

a

d

\*

+

c

Sin x

b

* 1. Viết lại biểu thức dưới dạng tiền tố, hậu tố.

🟅Biểu thức tiền tố:

+ ,\*, \*, +, Sinx, b, c, d, \*, a, b

🟅Biểu thức hậu tố:

Sinx b + c \* d \* a b \* +

* 1. Mô tả quá trình định giá biểu thức bằng ngăn xếp.

Định giá biểu thức: Sinx b + c \* d \* a b \* + bằng ngăn xếp

Sinx + b = A A \* c = B B \* d =C a\*b=D C+D=K

b

b c d a D

Sinx A B C C K

**Câu 6.(3 điểm)**

Cho biểu thức B = cos(3\*x) + a/m! - (d \* c – b )

-

- Dựng cây nhị phân biểu diễn biểu thức

-

+

cos

b

/

\*

\*

c

dc

!

a

x

3

m

- Viết lại biểu thức dưới dạng tiền tố, hậu tố

🟅Biểu thức tiền tố: (thăm gốc , duyệt cây con trái theo thứ tự trước, duyệt cây con phải theo thứ tự trước)

- + cos \* 3 x / a ! m - \* d c b

🟅Biểu thức hậu tố:

3 x \* cos a m ! / + d c \* b - -

- Minh hoạ việc định giá biểu thức bằng ngăn xếp với biểu thức cho dạng hậu tố.

🟅Định giá biêu thức: 3 x \* cos a m ! / + d c \* b - -

3\*x=A m! = B B + cosA =C d \* c = D

m

a B c

x cos cos d

3 A A C

D-b=E C–E=K

b

D E

C C K

**Câu 7.(3 điểm)**

Cho biểu thức B = x2 + (a\*cosx - d )\* c – b/a

- Dựng cây nhị phân biểu diễn biểu thức.

-

/

+

a

b

\*

\*

x

x

c

-

d

\*

a

cosx

- Viết lại biểu thức dưới dạng tiền tố, hậu tố.

🟅Biểu thức tiền tố:

- + \* xx \* - \* a cosx d c / b a

🟅Biểu thức hậu tố:

xx \* a cosx \* d – c \* + b a / -

- Minh hoạ việc định giá biểu thức bằng ngăn xếp với biểu thức cho dạng hậu tố.

🟅Định giá biểu thức hậu tố : xx \* a cosx \* d – c \* + b a / -

x \* x = A a \* cosx = B B – d = C C \* c =D

cos d c

x a B C

x A A A

A + D = E b / a = F E – F = K

a

D b F

A E E K