Bài 2:

Cây ban đầu:

2

/ \

1 10

/ \

6 13

/ \ \

3 8 20

/

7

Cây sau khi chèn:

2

/ \

1 10

/ / \

0 6 13

/ \ \

3 8 20

/ /

7 14

\

35

Cây sau khi xóa:

2

/ \

1 10

/ / \

0 7 14

/ \ \

3 8 20

Bài 3:

Cây ban đầu:

2

/ \

1 10

/ \ / \

6 3 8 7

/ \

13 20

Cây sau khi chèn:

0

/ \

1 10

/ \ / \

6 3 8 7

/ \ /

13 20 14

/

35

Cây sau khi xóa:

0

/ \

1 10

/ \ / \

7 3 8 14

/ \ /

20 35 10

Bài 4:

Tập hợp 10 số ngẫu nhiên từ 1-20

S1 = {15, 1, 12, 19, 2, 2, 7, 18, 13, 15}

Cây tìm kiếm nhị phân sau mỗi bước chèn:

19

18

15

13

12

7

2

1

Tìm và xóa phần tử lớn nhất:

Node\* findMax(Node\* root) {

if (root == nullptr) {

return nullptr;

} else if (root->right == nullptr) {

return root;

} else {

return findMax(root->right);

}

}

Node\* removeMax(Node\* root) {

if (root == nullptr) {

return nullptr;

} else if (root->right != nullptr) {

root->right = removeMax(root->right);

return root;

} else {

Node\* temp = root->left;

delete root;

return temp;

}

}

Cây sau khi xóa phần tử lớn nhất:

18

15

13

12

7

2

1

Tìm và xóa phần tủ nhỏ nhất trong cây:

Node\* removeMin(Node\* root) {

if (root == nullptr) {

return nullptr;

} else if (root->left != nullptr) {

root->left = removeMin(root->left);

return root;

} else {

Node\* temp = root->right;

delete root;

return temp;

}

}

Cây sau khi xóa phần tử nhỏ nhất:

18

15

13

12

7

Bài 5:

S2 = {5, 7, 3, 6, 20, 17, 10, 9, 4, 5}

S1 = {15, 1, 12, 19, 2, 2, 7, 18, 13, 15}

Cây nhị phân từ S2:

10

3

17

5

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 15:

10

3

17

5

15

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 1:

10

3

17

1

5

15

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 12:

10

3

12

17

1

5

15

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 19:

3

10

5

12

17

1

19

15

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 2:

2

3

10

5

12

17

1

19

15

20

5

7

4

6

9

Cây nhị phân sau khi chèn phần tử 2:

2

3

10

5

12

17

1

19

15

20

5

7

4

6

9

2

Cây nhị phân sau khi chèn phần tử 7:

2

3

10

5

12

17

1

19

15

20

5

7

4

6

7

9

2

Cây nhị phân sau khi chèn phần tử 18:

2

3

10

5

12

17

1

19

15

20

5

18

6

4

7

7

9

2

Cây nhị phân sau khi chèn phần tử 13:

2

3

10

5

12

17

1

19

15

20

5

18

6

7

4

13

7

9

2

Cây nhị phân sau khi chèn phần tử 15:

2

3

10

5

12

17

1

19

15

20

15

5

18

6

7

4

13

7

9

2

Tìm và xóa phần tử lớn nhất trong cây nhị phân:

int removeMax(vector<int>& arr, int n)

{

// Trường hợp cây rỗng

if (n == 0)

return -1;

// Lưu giá trị của phần tử lớn nhất

int maxElement = arr[0];

// Gán giá trị của phần tử cuối cùng cho phần tử đầu tiên

arr[0] = arr[n - 1];

// Giảm kích thước của cây đi 1

n--;

// Hàm để sắp xếp lại cây

int i = 0;

while (true)

{

int left = 2 \* i + 1;

int right = 2 \* i + 2;

int largest = i;

// So sánh với nút con bên trái

if (left < n && arr[left] > arr[largest])

largest = left;

// So sánh với nút con bên phải

if (right < n && arr[right] > arr[largest])

largest = right;

// Nếu nút hiện tại không phải là nút lớn nhất, hoán đổi giá trị và tiếp tục

if (largest != i)

{

swap(arr[i], arr[largest]);

i = largest;

}

else

{

break;

}

}

// Trả về giá trị của phần tử lớn nhất

return maxElement;

}