

Bài tiểu luận cấu trúc dữ liệu

Nhóm: 7

Võ Thương Trường Nhon – Phạm Đức Tài

Chủ đề 1: CẤU TRÚC DANH SÁCH:

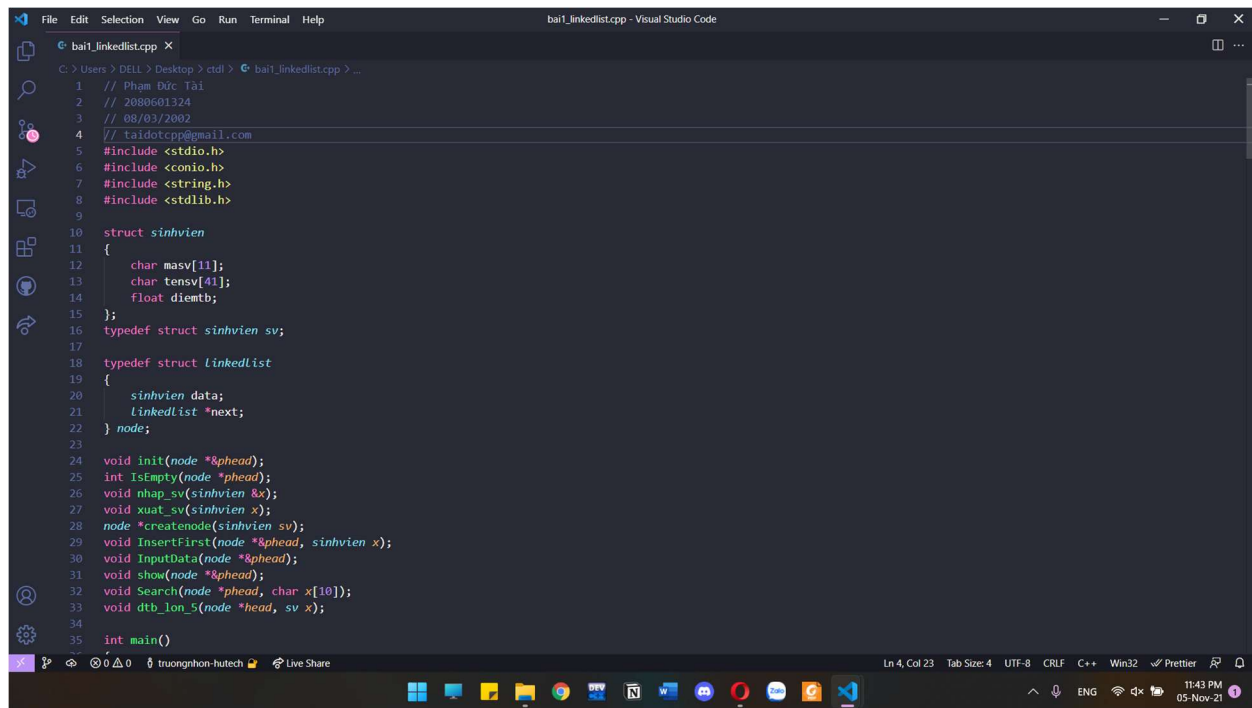
Câu 1: Viết chương trình quản lý danh sách sinh viên (sử dụng DSLKĐ), thông tin mỗi sv gồm:

Mã sv - chuỗi tối đa 10 kí tự, Họ tên - chuỗi tối đa 40 kí tự, Điểm trung bình - số thực.

Chương trình có các chức năng sau:

- Tạo 1 danh sách gồm n sv (n nhập từ bàn phím, thông tin của mỗi sv nhập từ bàn phím)
- Xuất danh sách sinh viên
- Xuất thông tin các sv có DTB>5
- Tìm sinh viên có tên là X

Bài làm:



```
1 // Phạm Đức Tài
2 // 2080601324
3 // 08/03/2002
4 // taidotcpp@gmail.com
5 #include <stdio.h>
6 #include <conio.h>
7 #include <string.h>
8 #include <stdlib.h>
9
10 struct sinhvien
11 {
12     char masv[11];
13     char tensv[41];
14     float diemtb;
15 };
16 typedef struct sinhvien sv;
17
18 typedef struct LinkedList
19 {
20     sinhvien data;
21     LinkedList *next;
22 } node;
23
24 void init(node *phead);
25 int IsEmpty(node *phead);
26 void nhap_sv(sinhvien &x);
27 void xuat_sv(sinhvien x);
28 node *createnode(sinhvien sv);
29 void InsertFirst(node *phead, sinhvien x);
30 void InputData(node *phead);
31 void show(node *phead);
32 void Search(node *phead, char x[10]);
33 void dtb_lon_5(node *head, sv x);
34
35 int main()
```

```
File Edit Selection View Go Run Terminal Help
bai1_linkedlist.cpp - ctdl - Visual Studio Code

bai1_linkedlist.cpp X
bai1_linkedlist.cpp > main()
34
35 int main()
36 {
37     node *head;
38     sv x;
39     init(head);
40     int n;
41     do
42     {
43         system("cls");
44         printf("CHUONG TRINH LINK-LIST DAY SO\n");
45         printf("\n*****");
46         printf("\n* 1. Nhap danh sach sinh vien *");
47         printf("\n* 2. Xuat danh sach sinh vien *");
48         printf("\n* 3. Xuat sv co dtb>5 *");
49         printf("\n* 4. Tim sinh vien co ma X *");
50         printf("\n* 0. Thoat chuong trinh *");
51         printf("\n*****");
52
53         printf("\nChon chuc nang n: ");
54         scanf("%d", &n);
55         switch (n)
56         {
57             case 1:
58             {
59                 printf("\nNhap danh sach sinh vien:\n");
60                 InputData(head);
61                 printf("\nNhan phim bat ki de tiep tucl");
62                 getch();
63                 break;
64             }
65             case 2:
66             {
67                 printf("\n Xuat danh sach sinh vien: ");
68                 show(head);
```

```
File Edit Selection View Go Run Terminal Help
bai1_linkedlist.cpp - ctdl - Visual Studio Code

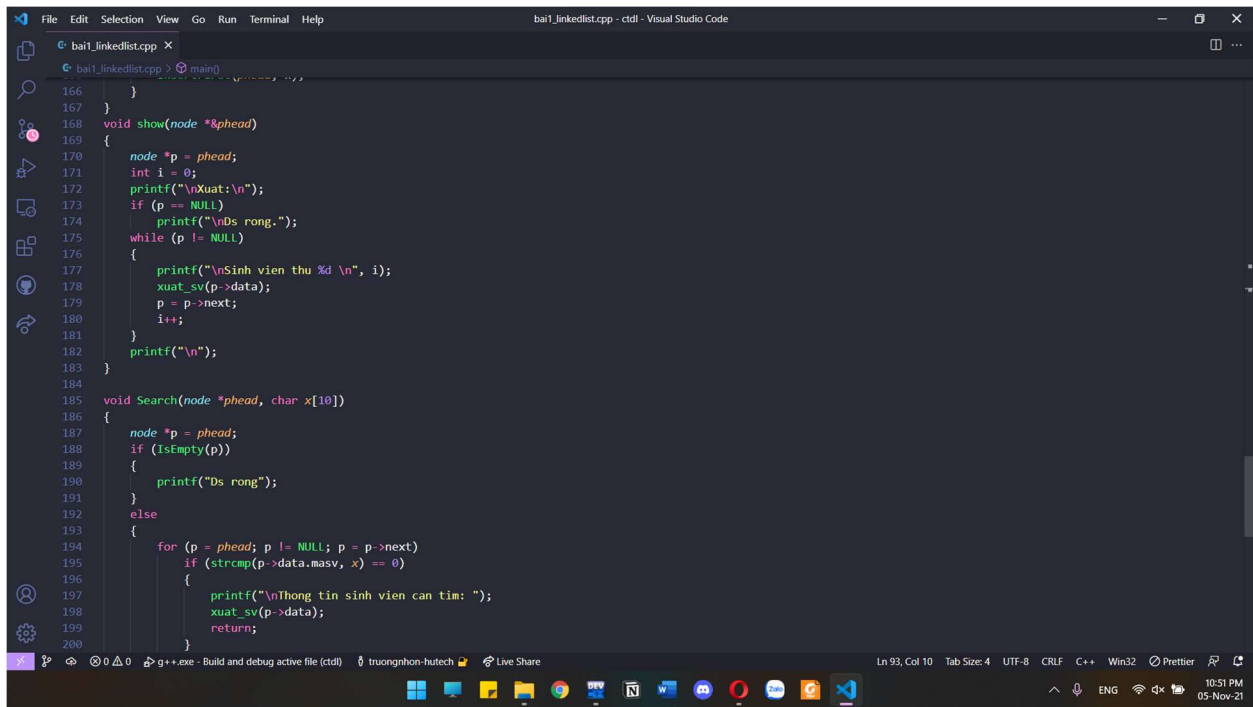
bai1_linkedlist.cpp X
bai1_linkedlist.cpp > main()
65
66     {
67         printf("\n Xuat danh sach sinh vien: ");
68         show(head);
69         printf("\nNhan phim bat ki de tiep tucl");
70         getch();
71         break;
72     }
73     case 3:
74     {
75         sv cau3;
76         printf("\nDs sinh vien co dtb > 5: ");
77         dtb_lon_5(head, cau3);
78         printf("\nNhan phim bat ki de tiep tucl");
79         getch();
80         break;
81     }
82     case 4:
83     {
84         char cau4[10];
85         printf("\nNhap mssv can tim: ");
86         fflush(stdin);
87         gets(cau4);
88         printf("\nSinh vien can tim la: \n");
89         Search(head, cau4);
90         printf("\nNhan phim bat ki de tiep tucl");
91         getch();
92         break;
93     }
94     case 0:
95     {
96         printf("\n");
97         printf("\n Cam on da su dung chuong trinh cua chung toi!");
98         getch();
99         return 0;
100     }
}
```

```
File Edit Selection View Go Run Terminal Help
bai1_linkedlist.cpp - ctdl - Visual Studio Code

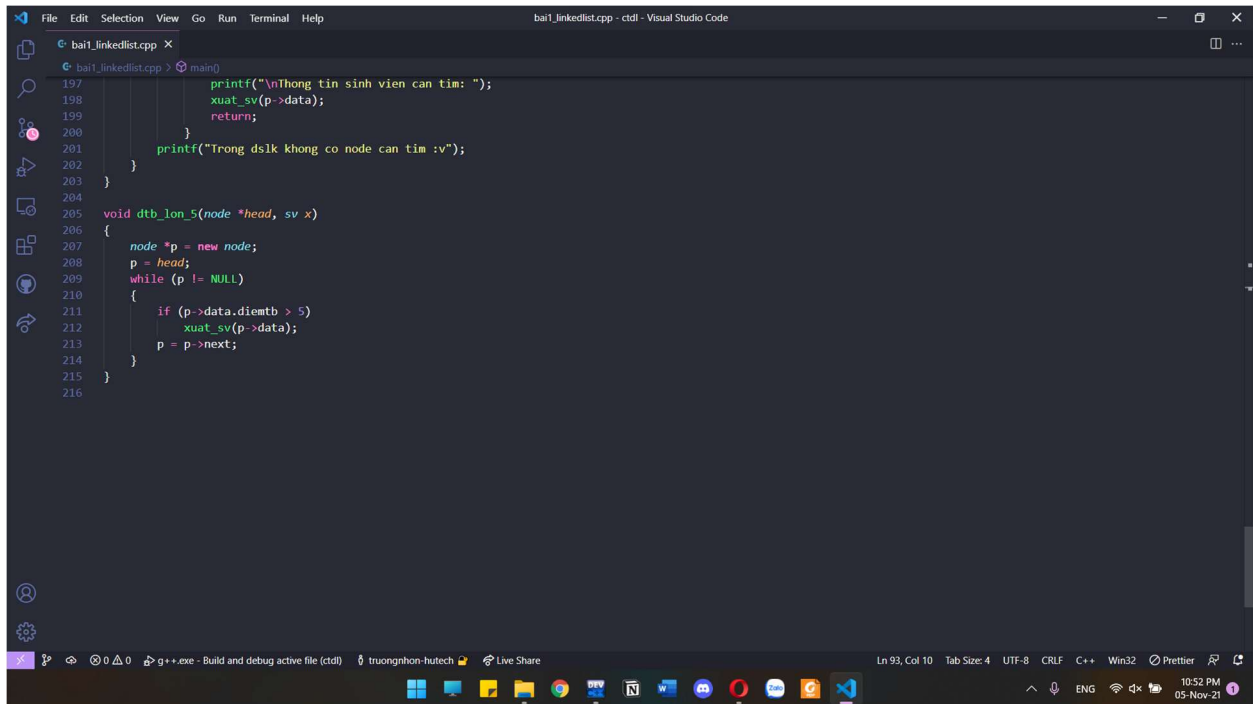
bai1_linkedlist.cpp X
bai1_linkedlist.cpp > main()
100
101     default:
102         printf("Lua chon trong 0-4! ");
103         printf("\nNhan phim bat ki de tiep tuc!");
104         getch();
105     }
106     while (n != 0);
107     return 0;
108 }
109 void init(node *&phead)
110 {
111     phead = NULL;
112 }
113 int isEmpty(node *&phead)
114 {
115     return phead == NULL;
116 }
117
118 void nhap_sv(sinhvien &x)
119 {
120     printf("Nhap ma so sv: ");
121     fflush(stdin);
122     gets(x.masv);
123     printf("Nhap ten sv: ");
124     fflush(stdin);
125     gets(x.tensv);
126     printf("Nhap diem trung binh: ");
127     scanf("%f", &x.diemtb);
128     printf("\n\n");
129 }
130 void xuat_sv(sinhvien x)
131 {
132     printf("%s-%s-%.2f\n", x.masv, x.tensv, x.diemtb);
133 }
134
g++ .exe - Build and debug active file (ctdl) truongnhon-hutech Live Share Ln 93, Col 10 Tab Size: 4 UTF-8 CRLF C++ Win32 Prettier 10:41 PM 05-Nov-21
```

```
File Edit Selection View Go Run Terminal Help
bai1_linkedlist.cpp - ctdl - Visual Studio Code

bai1_linkedlist.cpp X
bai1_linkedlist.cpp > main()
132     printf("%s-%s-%.2f\n", x.masv, x.tensv, x.diemtb);
133 }
134
135 node *createnode(sinhvien sv)
136 {
137     node *p = new node;
138     p->next = NULL;
139     p->data = sv;
140     return p;
141 }
142
143 void InsertFirst(node *&phead, sinhvien x)
144 {
145     node *p = createnode(x);
146
147     if (isEmpty(phead))
148         phead = p;
149     else
150     {
151         p->next = phead;
152         phead = p;
153     }
154 }
155 void InputData(node *&phead)
156 {
157     int n;
158     printf("Nhap n: ");
159     scanf("%d", &n);
160     sinhvien x;
161     for (int i = 0; i < n; i++)
162     {
163         printf("Nhap sinh vien thu %d: \n", i);
164         nhap_sv(x);
165         InsertFirst(phead, x);
166     }
167 }
```



```
166     }
167 }
168 void show(node *&phead)
169 {
170     node *p = phead;
171     int i = 0;
172     printf("\nXuat:\n");
173     if (p == NULL)
174         printf("\nDs rong.");
175     while (p != NULL)
176     {
177         printf("\nSinh vien thu %d \n", i);
178         xuat_sv(p->data);
179         p = p->next;
180         i++;
181     }
182     printf("\n");
183 }
184
185 void Search(node *phead, char x[10])
186 {
187     node *p = phead;
188     if (IsEmpty(p))
189     {
190         printf("Ds rong");
191     }
192     else
193     {
194         for (p = phead; p != NULL; p = p->next)
195             if (strcmp(p->data.masv, x) == 0)
196             {
197                 printf("\nThong tin sinh vien can tim: ");
198                 xuat_sv(p->data);
199                 return;
200             }
```



```
197         printf("\nThong tin sinh vien can tim: ");
198         xuat_sv(p->data);
199         return;
200     }
201     printf("Trong dsdk khong co node can tim :v");
202 }
203 }
204
205 void dtb_lon_5(node *head, sv x)
206 {
207     node *p = new node;
208     p = head;
209     while (p != NULL)
210     {
211         if (p->data.diemtb > 5)
212             xuat_sv(p->data);
213         p = p->next;
214     }
215 }
216
```

Chủ đề 2: CẤU TRÚC STACK & QUEUE

Câu 1: Viết chương trình cài đặt bài toán chuyển biểu thức trung tố sang hậu tố, sau đó tính giá trị biểu thức hậu tố.

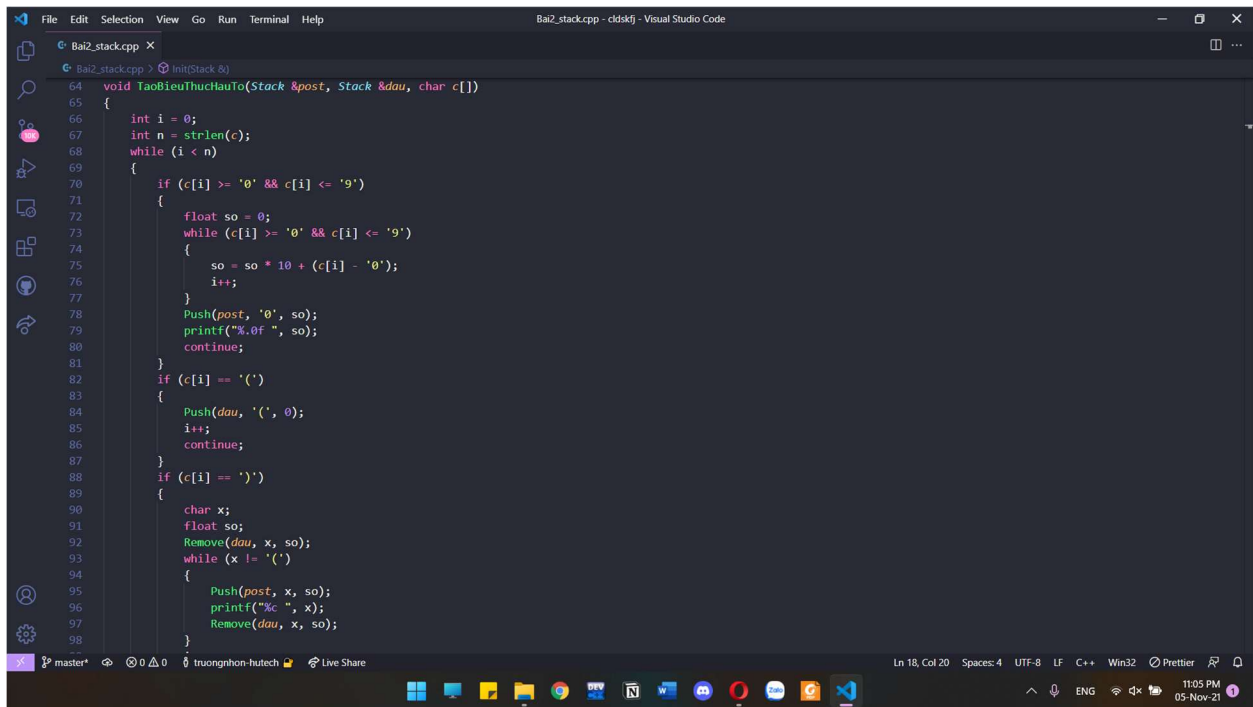
Bài làm:

```
File Edit Selection View Go Run Terminal Help
Bai2_stack.cpp - cldskj - Visual Studio Code

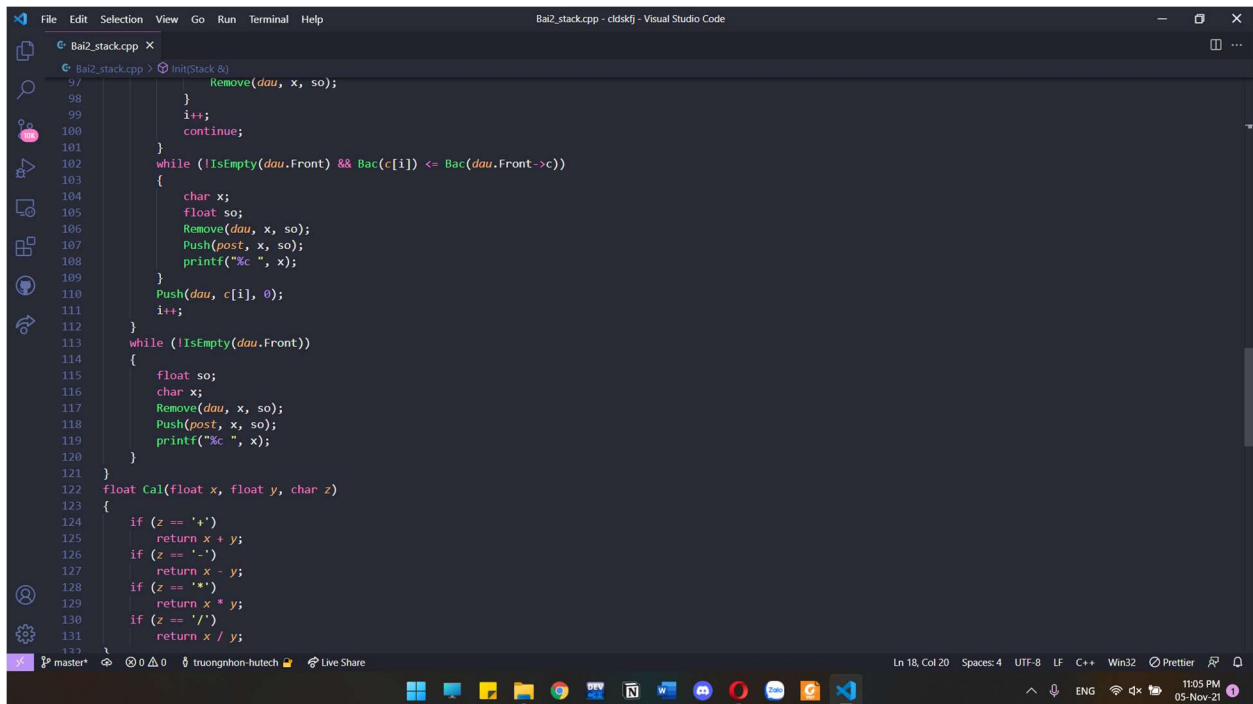
C: Bai2_stack.cpp X
C: Bai2_stack.cpp > ...
1 // Phạm Đức Tài
2 // 2080601324
3 // 08/03/2002
4 // taidotcpp@gmail.com
5 #include <stdio.h>
6 #include <string.h>
7
8 struct Node
9 {
10     float so;
11     char c;
12     Node *next;
13 };
14 struct Stack
15 {
16     Node *Front;
17 };
18 void Init(Stack &st)
19 {
20     st.Front = NULL;
21 }
22 int IsEmpty(Node *x)
23 {
24     return x == NULL;
25 }
26
27 Node *CreateNode(char x, float so)
28 {
29     Node *p = new Node;
30     p->c = x;
31     p->so = so;
32     p->next = NULL;
33     return p;
34 }
35 void Push(Stack &st, char x, float so)
```

```
File Edit Selection View Go Run Terminal Help
Bai2_stack.cpp - cldskj - Visual Studio Code

C: Bai2_stack.cpp X
C: Bai2_stack.cpp > Init(Stack &st)
35 void Push(Stack &st, char x, float so)
36 {
37     Node *p = CreateNode(x, so);
38     if (IsEmpty(st.Front))
39     {
40         st.Front = p;
41         return;
42     }
43     p->next = st.Front;
44     st.Front = p;
45 }
46 void Remove(Stack &st, char &x, float &so)
47 {
48     if (IsEmpty(st.Front))
49         return;
50     Node *p = st.Front;
51     st.Front = p->next;
52     x = p->c;
53     so = p->so;
54     delete p;
55 }
56 int Bac(char x)
57 {
58     if (x == '(')
59         return 0;
60     if (x == '+' || x == '-')
61         return 1;
62     return 2;
63 }
64 void TaoBieuThucHauTo(Stack &post, Stack &dau, char c[])
65 {
66     int i = 0;
67     int n = strlen(c);
68     while (i < n)
69     {
```



```
void TaoBieuThuchauTo(Stack &post, Stack &dau, char c[])
{
    int i = 0;
    int n = strlen(c);
    while (i < n)
    {
        if (c[i] >= '0' && c[i] <= '9')
        {
            float so = 0;
            while (c[i] >= '0' && c[i] <= '9')
            {
                so = so * 10 + (c[i] - '0');
                i++;
            }
            Push(post, '0', so);
            printf("%.0f ", so);
            continue;
        }
        if (c[i] == '(')
        {
            Push(dau, '(', 0);
            i++;
            continue;
        }
        if (c[i] == ')')
        {
            char x;
            float so;
            Remove(dau, x, so);
            while (x != '(')
            {
                Push(post, x, so);
                printf("%c ", x);
                Remove(dau, x, so);
            }
        }
    }
}
```



```
Remove(dau, x, so);
}
i++;
continue;
}
while (!IsEmpty(dau.Front) && Bac(c[i]) <= Bac(dau.Front->c))
{
    char x;
    float so;
    Remove(dau, x, so);
    Push(post, x, so);
    printf("%c ", x);
}
Push(dau, c[i], 0);
i++;
}
while (!IsEmpty(dau.Front))
{
    float so;
    char x;
    Remove(dau, x, so);
    Push(post, x, so);
    printf("%c ", x);
}
}
float Cal(float x, float y, char z)
{
    if (z == '+')
        return x + y;
    if (z == '-')
        return x - y;
    if (z == '*')
        return x * y;
    if (z == '/')
        return x / y;
}
```

```
File Edit Selection View Go Run Terminal Help
Bai2_stack.cpp - ctdskj - Visual Studio Code

Bai2_stack.cpp X
Bai2_stack.cpp > Init(Stack &S)
131     return x / y;
132 }
133 float GiaTriBieuThuc(Stack &post, Stack &dau)
134 {
135     char x;
136     float so;
137     while (!IsEmpty(post.Front))
138     {
139         Remove(post, x, so);
140         if (x == '0')
141         {
142             while (!IsEmpty(dau.Front) && dau.Front->c == '0')
143             {
144                 char x1, x2;
145                 float so1, so2;
146                 Remove(dau, x1, so1);
147                 Remove(dau, x2, so2);
148                 so = Cal(so, so1, x2);
149             }
150             Push(dau, x, so);
151         }
152         Remove(dau, x, so);
153     }
154     return so;
155 }
156 int main()
157 {
158     char c[1000];
159     fflush(stdin);
160     printf("Nhap bieu thuc can bien doi: ");
161     scanf("%s", &c);
162     Stack post, dau;
163     Init(post);
164     Init(dau);
165     printf("Bieu thuc hau to: ");
166 }
master* 0 0 0 truongnhon-hutech Live Share
Ln 18, Col 20 Spaces: 4 UTF-8 LF C++ Win32 Prettier 11:06 PM 05-Nov-21
```

```
File Edit Selection View Go Run Terminal Help
Bai2_stack.cpp - ctdskj - Visual Studio Code

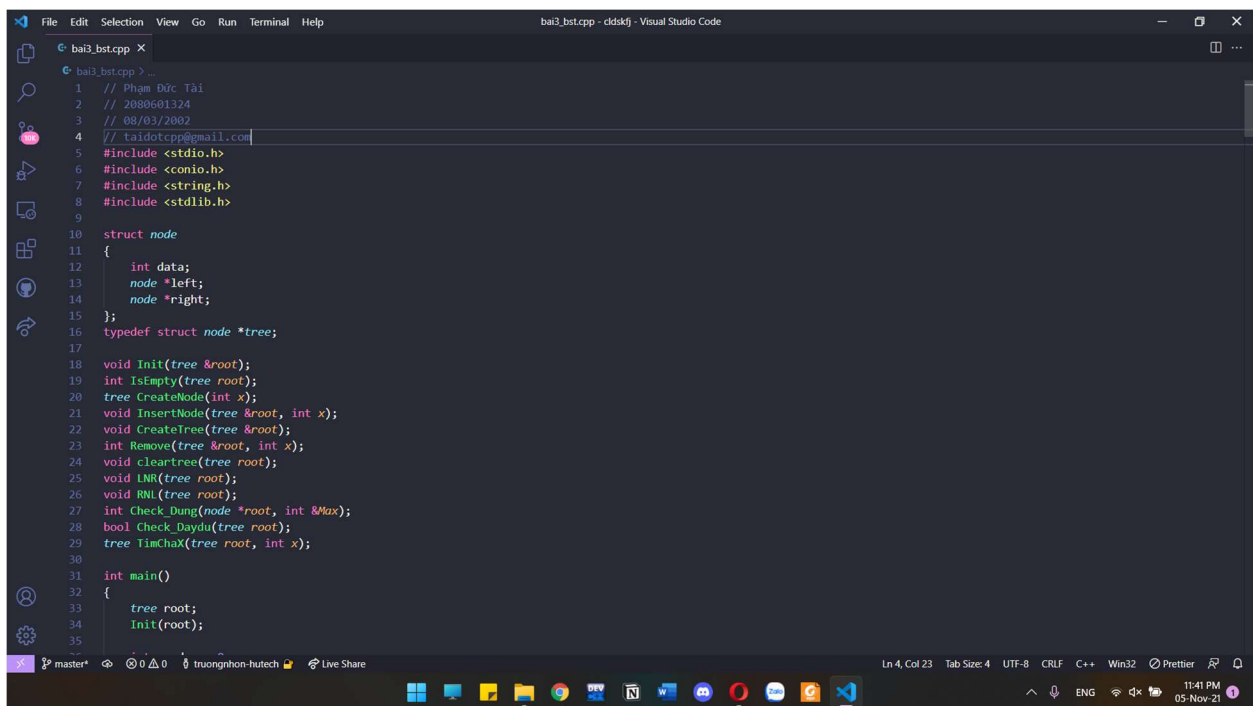
Bai2_stack.cpp X
Bai2_stack.cpp > Init(Stack &S)
148     so = Cal(so, so1, x2);
149 }
150 }
151 Push(dau, x, so);
152 }
153 Remove(dau, x, so);
154 return so;
155 }
156 int main()
157 {
158     char c[1000];
159     fflush(stdin);
160     printf("Nhap bieu thuc can bien doi: ");
161     scanf("%s", &c);
162     Stack post, dau;
163     Init(post);
164     Init(dau);
165     printf("Bieu thuc hau to: ");
166     TaoBieuThucHauTo(post, dau, c);
167     printf("\n");
168     float t = GiaTriBieuThuc(post, dau);
169     printf("%.02f", t);
170 }
171
master* 0 0 0 truongnhon-hutech Live Share
Ln 18, Col 20 Spaces: 4 UTF-8 LF C++ Win32 Prettier 11:06 PM 05-Nov-21
```


Chủ đề 3: CẤU TRÚC CÂY - CÂY NHỊ PHÂN – CÂY NHỊ PHÂN TÌM KIẾM

Câu 1: Cài đặt cấu trúc dữ liệu liên kết cho cây nhị phân tìm kiếm, với các thao tác:

- a) Cài đặt các thao tác xây dựng cây: Init, IsEmpty, CreateNode
- b) Cài đặt thao tác cập nhật: Insert, Remove, ClearTree
- c) Xuất danh sách tăng dần và giảm dần
- d) Kiểm tra xem cây có phải là cây nhị phân đúng
- e) Kiểm tra xem cây có phải là cây nhị phân đầy đủ
- f) Xác định nút cha của nút chứa khoá X

Bài làm:



```
1 // Phạm Đức Tài
2 // 2080601324
3 // 08/03/2002
4 // taidotcpp@gmail.com
5 #include <stdio.h>
6 #include <conio.h>
7 #include <string.h>
8 #include <stdlib.h>
9
10 struct node
11 {
12     int data;
13     node *left;
14     node *right;
15 };
16 typedef struct node *tree;
17
18 void Init(tree &root);
19 int IsEmpty(tree root);
20 tree CreateNode(int x);
21 void InsertNode(tree &root, int x);
22 void CreateTree(tree &root);
23 int Remove(tree &root, int x);
24 void clearTree(tree root);
25 void LNR(tree root);
26 void RNL(tree root);
27 int Check_Dung(node *root, int &Max);
28 bool Check_Daydu(tree root);
29 tree TimChaX(tree root, int x);
30
31 int main()
32 {
33     tree root;
34     Init(root);
35 }
```



```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code
bai3_bst.cpp
36 int n, dem = 0;
37 do
38 {
39     system("cls");
40     printf("CHUONG TRINH CAY NHI PHAN TIM KIEM\n");
41     printf("\n*****");
42     printf("\n* 1/ Nhap du lieu cho cay nhi phan *");
43     printf("\n* 2/ Xuat cay nhi phan tang dan *");
44     printf("\n* 3/ Xuat cay nhi phan giam dan *");
45     printf("\n* 4/ Xoa node x tren cay *");
46     printf("\n* 5/ Xoa toan bo cay *");
47     printf("\n* 6/ Kiem tra xem co phai cay nhi phan day du *");
48     printf("\n* 7/ Kiem tra xem co phai cay nhi phan dung *");
49     printf("\n* 8/ Tim node cha cua node x *");
50     printf("\n* 0/ Thoat chuong trinh *");
51     printf("\n*****");
52
53     printf("\nChon chuc nang n: ");
54     scanf("%d", &n);
55     switch (n)
56     {
57     case 1:
58     {
59         printf("\n");
60         printf("Nhap du lieu cho cay nhi phan\n");
61         CreateTree(root);
62         printf("\nNhap phim bat ky de tiep tuc!!!");
63         getch();
64         break;
65     }
66     case 2:
67     {
68         printf("\nXuat cay nhi phan tang dan\n");
69         LNR(root);
70         printf("\nNhap phim bat ky de tiep tuc!!!");
71     }
72     }
73 }
74 while (n != 0);
```

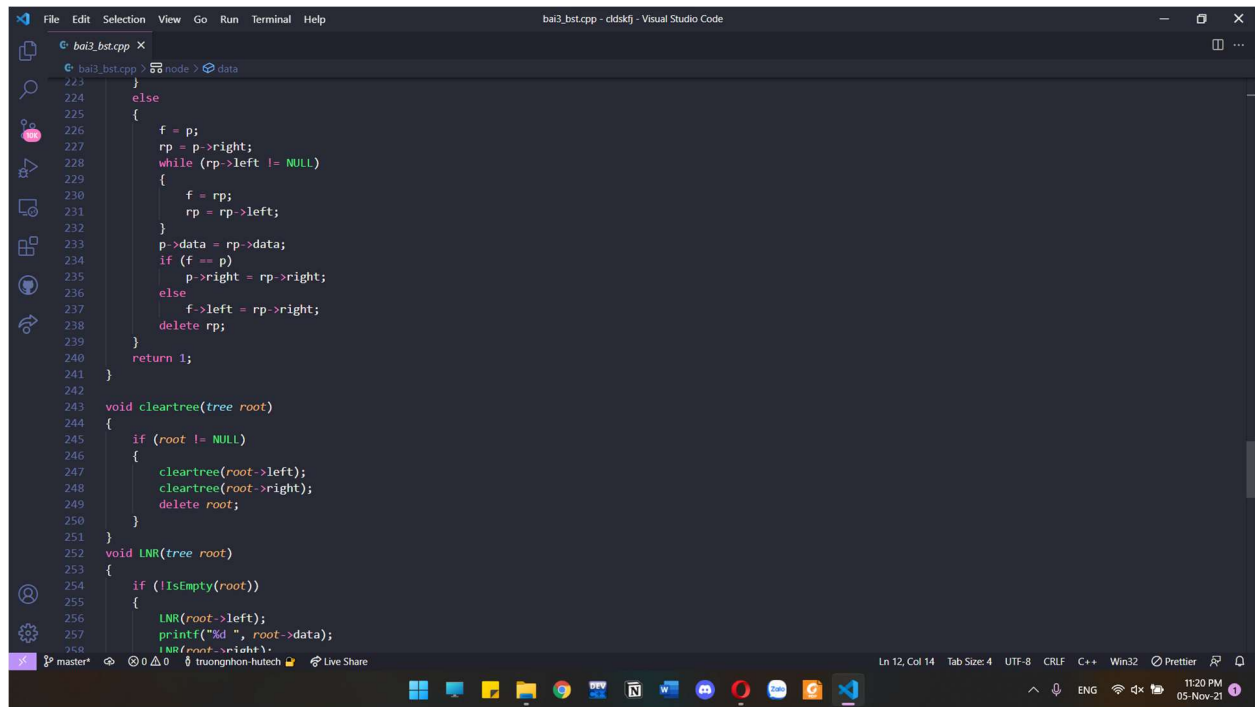
```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code
bai3_bst.cpp
67 {
68     printf("\nXuat cay nhi phan tang dan\n");
69     LNR(root);
70     printf("\nNhap phim bat ky de tiep tuc!!!");
71     getch();
72     break;
73 }
74 case 3:
75 {
76     printf("\n 3/ Xuat cay nhi phan giam dan\n");
77     RNL(root);
78     printf("\nNhap phim bat ky de tiep tuc!!!");
79     getch();
80     break;
81 }
82 case 4:
83 {
84     printf("\n 4/ Xoa node x tren cay nhi phan\n");
85     int cau4;
86     printf("\nNhap nut x can xoa: ");
87     scanf("%d", &cau4);
88     if (Remove(root, cau4) == 1)
89         printf("\nDa xoa thanh cong");
90     else
91         printf("\nChua xoa duoc");
92     printf("\nNhap phim bat ky de tiep tuc!!!");
93     getch();
94     break;
95 }
96 case 5:
97 {
98     printf("\n5/ Xoa toan bo cay: \n");
99     cleartree(root);
100     printf("\nNhap phim bat ky de tiep tuc!!!");
101     getch();
102 }
```

```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskfj - Visual Studio Code
bai3_bst.cpp
node > data
99      clearTree(root);
100      printf("\nNhap phim bat ky de tiep tuc!!!");
101      getch();
102      break;
103  }
104  case 6:
105  {
106      printf("\n6/ Kiem tra xem co phai cay nhi phan day du\n");
107      if (check_Daydu(root))
108          printf("\nDung la cay nhi phan DAY DU");
109      else
110          printf("\nKhong phai cay nhi phan DAY DU");
111      printf("\nNhap phim bat ky de tiep tuc!!!");
112      getch();
113      break;
114  }
115  case 7:
116  {
117      printf("\n7/ Kiem tra xem co phai cay nhi phan dung\n");
118      int max = -10000;
119      if (check_Dung(root, max))
120          printf("\nLa cay nhi phan dung.");
121      else
122          printf("\nLa cay nhi phan dung");
123      printf("\nNhap phim bat ky de tiep tuc!!!");
124      getch();
125      break;
126  }
127  case 8:
128  {
129      printf("\n8/ Tim node cha cua node x\n");
130      int cau8;
131      printf("\nNhap gia tri node x: ");
132      scanf("%d", &cau8);
133      tree p = TimChaX(root, cau8);
134  }
135  }
136  }
137  }
138  }
139  }
140  }
141  }
142  }
143  }
144  }
145  }
146  }
147  }
148  }
149  }
150  }
151  }
152  }
153  }
154  }
155  }
156  }
157  }
158  }
159  }
160  }
161  }
162  }
163  }
164  }
165  }
166  }
167  }
168  }
169  }
170  }
171  }
172  }
173  }
174  }
175  }
176  }
177  }
178  }
179  }
180  }
181  }
182  }
183  }
184  }
185  }
186  }
187  }
188  }
189  }
190  }
191  }
192  }
193  }
194  }
195  }
196  }
197  }
198  }
199  }
200  }
201  }
202  }
203  }
204  }
205  }
206  }
207  }
208  }
209  }
210  }
211  }
212  }
213  }
214  }
215  }
216  }
217  }
218  }
219  }
220  }
221  }
222  }
223  }
224  }
225  }
226  }
227  }
228  }
229  }
230  }
231  }
232  }
233  }
234  }
235  }
236  }
237  }
238  }
239  }
240  }
241  }
242  }
243  }
244  }
245  }
246  }
247  }
248  }
249  }
250  }
251  }
252  }
253  }
254  }
255  }
256  }
257  }
258  }
259  }
260  }
261  }
262  }
263  }
264  }
265  }
266  }
267  }
268  }
269  }
270  }
271  }
272  }
273  }
274  }
275  }
276  }
277  }
278  }
279  }
280  }
281  }
282  }
283  }
284  }
285  }
286  }
287  }
288  }
289  }
290  }
291  }
292  }
293  }
294  }
295  }
296  }
297  }
298  }
299  }
300  }
301  }
302  }
303  }
304  }
305  }
306  }
307  }
308  }
309  }
310  }
311  }
312  }
313  }
314  }
315  }
316  }
317  }
318  }
319  }
320  }
321  }
322  }
323  }
324  }
325  }
326  }
327  }
328  }
329  }
330  }
331  }
332  }
333  }
334  }
335  }
336  }
337  }
338  }
339  }
340  }
341  }
342  }
343  }
344  }
345  }
346  }
347  }
348  }
349  }
350  }
351  }
352  }
353  }
354  }
355  }
356  }
357  }
358  }
359  }
360  }
361  }
362  }
363  }
364  }
365  }
366  }
367  }
368  }
369  }
370  }
371  }
372  }
373  }
374  }
375  }
376  }
377  }
378  }
379  }
380  }
381  }
382  }
383  }
384  }
385  }
386  }
387  }
388  }
389  }
390  }
391  }
392  }
393  }
394  }
395  }
396  }
397  }
398  }
399  }
400  }
401  }
402  }
403  }
404  }
405  }
406  }
407  }
408  }
409  }
410  }
411  }
412  }
413  }
414  }
415  }
416  }
417  }
418  }
419  }
420  }
421  }
422  }
423  }
424  }
425  }
426  }
427  }
428  }
429  }
430  }
431  }
432  }
433  }
434  }
435  }
436  }
437  }
438  }
439  }
440  }
441  }
442  }
443  }
444  }
445  }
446  }
447  }
448  }
449  }
450  }
451  }
452  }
453  }
454  }
455  }
456  }
457  }
458  }
459  }
460  }
461  }
462  }
463  }
464  }
465  }
466  }
467  }
468  }
469  }
470  }
471  }
472  }
473  }
474  }
475  }
476  }
477  }
478  }
479  }
480  }
481  }
482  }
483  }
484  }
485  }
486  }
487  }
488  }
489  }
490  }
491  }
492  }
493  }
494  }
495  }
496  }
497  }
498  }
499  }
500  }
501  }
502  }
503  }
504  }
505  }
506  }
507  }
508  }
509  }
510  }
511  }
512  }
513  }
514  }
515  }
516  }
517  }
518  }
519  }
520  }
521  }
522  }
523  }
524  }
525  }
526  }
527  }
528  }
529  }
530  }
531  }
532  }
533  }
534  }
535  }
536  }
537  }
538  }
539  }
540  }
541  }
542  }
543  }
544  }
545  }
546  }
547  }
548  }
549  }
550  }
551  }
552  }
553  }
554  }
555  }
556  }
557  }
558  }
559  }
560  }
561  }
562  }
563  }
564  }
565  }
566  }
567  }
568  }
569  }
570  }
571  }
572  }
573  }
574  }
575  }
576  }
577  }
578  }
579  }
580  }
581  }
582  }
583  }
584  }
585  }
586  }
587  }
588  }
589  }
590  }
591  }
592  }
593  }
594  }
595  }
596  }
597  }
598  }
599  }
600  }
601  }
602  }
603  }
604  }
605  }
606  }
607  }
608  }
609  }
610  }
611  }
612  }
613  }
614  }
615  }
616  }
617  }
618  }
619  }
620  }
621  }
622  }
623  }
624  }
625  }
626  }
627  }
628  }
629  }
630  }
631  }
632  }
633  }
634  }
635  }
636  }
637  }
638  }
639  }
640  }
641  }
642  }
643  }
644  }
645  }
646  }
647  }
648  }
649  }
650  }
651  }
652  }
653  }
654  }
655  }
656  }
657  }
658  }
659  }
660  }
661  }
662  }
663  }
664  }
665  }
666  }
667  }
668  }
669  }
670  }
671  }
672  }
673  }
674  }
675  }
676  }
677  }
678  }
679  }
680  }
681  }
682  }
683  }
684  }
685  }
686  }
687  }
688  }
689  }
690  }
691  }
692  }
693  }
694  }
695  }
696  }
697  }
698  }
699  }
700  }
701  }
702  }
703  }
704  }
705  }
706  }
707  }
708  }
709  }
710  }
711  }
712  }
713  }
714  }
715  }
716  }
717  }
718  }
719  }
720  }
721  }
722  }
723  }
724  }
725  }
726  }
727  }
728  }
729  }
730  }
731  }
732  }
733  }
734  }
735  }
736  }
737  }
738  }
739  }
740  }
741  }
742  }
743  }
744  }
745  }
746  }
747  }
748  }
749  }
750  }
751  }
752  }
753  }
754  }
755  }
756  }
757  }
758  }
759  }
760  }
761  }
762  }
763  }
764  }
765  }
766  }
767  }
768  }
769  }
770  }
771  }
772  }
773  }
774  }
775  }
776  }
777  }
778  }
779  }
780  }
781  }
782  }
783  }
784  }
785  }
786  }
787  }
788  }
789  }
790  }
791  }
792  }
793  }
794  }
795  }
796  }
797  }
798  }
799  }
800  }
801  }
802  }
803  }
804  }
805  }
806  }
807  }
808  }
809  }
810  }
811  }
812  }
813  }
814  }
815  }
816  }
817  }
818  }
819  }
820  }
821  }
822  }
823  }
824  }
825  }
826  }
827  }
828  }
829  }
830  }
831  }
832  }
833  }
834  }
835  }
836  }
837  }
838  }
839  }
840  }
841  }
842  }
843  }
844  }
845  }
846  }
847  }
848  }
849  }
850  }
851  }
852  }
853  }
854  }
855  }
856  }
857  }
858  }
859  }
860  }
861  }
862  }
863  }
864  }
865  }
866  }
867  }
868  }
869  }
870  }
871  }
872  }
873  }
874  }
875  }
876  }
877  }
878  }
879  }
880  }
881  }
882  }
883  }
884  }
885  }
886  }
887  }
888  }
889  }
890  }
891  }
892  }
893  }
894  }
895  }
896  }
897  }
898  }
899  }
900  }
901  }
902  }
903  }
904  }
905  }
906  }
907  }
908  }
909  }
910  }
911  }
912  }
913  }
914  }
915  }
916  }
917  }
918  }
919  }
920  }
921  }
922  }
923  }
924  }
925  }
926  }
927  }
928  }
929  }
930  }
931  }
932  }
933  }
934  }
935  }
936  }
937  }
938  }
939  }
940  }
941  }
942  }
943  }
944  }
945  }
946  }
947  }
948  }
949  }
950  }
951  }
952  }
953  }
954  }
955  }
956  }
957  }
958  }
959  }
960  }
961  }
962  }
963  }
964  }
965  }
966  }
967  }
968  }
969  }
970  }
971  }
972  }
973  }
974  }
975  }
976  }
977  }
978  }
979  }
980  }
981  }
982  }
983  }
984  }
985  }
986  }
987  }
988  }
989  }
990  }
991  }
992  }
993  }
994  }
995  }
996  }
997  }
998  }
999  }
1000 }
```

```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskfj - Visual Studio Code
bai3_bst.cpp
node > data
131      printf("\nNhap gia tri node x: ");
132      scanf("%d", &cau8);
133      tree p = TimChaX(root, cau8);
134      if (p == NULL)
135          printf("\nNode khong co cha.");
136      else
137          printf("\nCha cua node %d la: %d", cau8, p->data);
138      printf("\nNhap phim bat ky de tiep tuc!!!");
139      getch();
140      break;
141  }
142  }
143  case 0:
144  {
145      printf("\n Cam on da su dung chuong trinh cua chung toi!");
146      printf("\nNhap phim bat ky de tiep tuc!!!");
147      getch();
148      return 0;
149  }
150  default:
151      printf("Lua chon trong 0-9!! ");
152      printf("\nNhap phim bat ki de tiep tuc!");
153      getch();
154  }
155  } while (n != 0);
156  return 0;
157  }
158  }
159  void Init(tree &root)
160  {
161      root = NULL;
162  }
163  int IsEmpty(tree root)
164  {
165      return root == NULL;
166  }
167  }
168  }
169  }
170  }
171  }
172  }
173  }
174  }
175  }
176  }
177  }
178  }
179  }
180  }
181  }
182  }
183  }
184  }
185  }
186  }
187  }
188  }
189  }
190  }
191  }
192  }
193  }
194  }
195  }
196  }
197  }
198  }
199  }
200  }
201  }
202  }
203  }
204  }
205  }
206  }
207  }
208  }
209  }
210  }
211  }
212  }
213  }
214  }
215  }
216  }
217  }
218  }
219  }
220  }
221  }
222  }
223  }
224  }
225  }
226  }
227  }
228  }
229  }
230  }
231  }
232  }
233  }
234  }
235  }
236  }
237  }
238  }
239  }
240  }
241  }
242  }
243  }
244  }
245  }
246  }
247  }
248  }
249  }
250  }
251  }
252  }
253  }
254  }
255  }
256  }
257  }
258  }
259  }
260  }
261  }
262  }
263  }
264  }
265  }
266  }
267  }
268  }
269  }
270  }
271  }
272  }
273  }
274  }
275  }
276  }
277  }
278  }
279  }
280  }
281  }
282  }
283  }
284  }
285  }
286  }
287  }
288  }
289  }
290  }
291  }
292  }
293  }
294  }
295  }
296  }
297  }
298  }
299  }
300  }
301  }
302  }
303  }
304  }
305  }
306  }
307  }
308  }
309  }
310  }
311  }
312  }
313  }
314  }
315  }
316  }
317  }
318  }
319  }
320  }
321  }
322  }
323  }
324  }
325  }
326  }
327  }
328  }
329  }
330  }
331  }
332  }
333  }
334  }
335  }
336  }
337  }
338  }
339  }
340  }
341  }
342  }
343  }
344  }
345  }
346  }
347  }
348  }
349  }
350  }
351  }
352  }
353  }
354  }
355  }
356  }
357  }
358  }
359  }
360  }
361  }
362  }
363  }
364  }
365  }
366  }
367  }
368  }
369  }
370  }
371  }
372  }
373  }
374  }
375  }
376  }
377  }
378  }
379  }
380  }
381  }
382  }
383  }
384  }
385  }
386  }
387  }
388  }
389  }
390  }
391  }
392  }
393  }
394  }
395  }
396  }
397  }
398  }
399  }
400  }
401  }
402  }
403  }
404  }
405  }
406  }
407  }
408  }
409  }
410  }
411  }
412  }
413  }
414  }
415  }
416  }
417  }
418  }
419  }
420  }
421  }
422  }
423  }
424  }
425  }
426  }
427  }
428  }
429  }
430  }
431  }
432  }
433  }
434  }
435  }
436  }
437  }
438  }
439  }
440  }
441  }
442  }
443  }
444  }
445  }
446  }
447  }
448  }
449  }
450  }
451  }
452  }
453  }
454  }
455  }
456  }
457  }
458  }
459  }
460  }
461  }
462  }
463  }
464  }
465  }
466  }
467  }
468  }
469  }
470  }
471  }
472  }
473  }
474  }
475  }
476  }
477  }
478  }
479  }
480  }
481  }
482  }
483  }
484  }
485  }
486  }
487  }
488  }
489  }
490  }
491  }
492  }
493  }
494  }
495  }
496  }
497  }
498  }
499  }
500  }
501  }
502  }
503  }
504  }
505  }
506  }
507  }
508  }
509  }
510  }
511  }
512  }
513  }
514  }
515  }
516  }
517  }
518  }
519  }
520  }
521  }
522  }
523  }
524  }
525  }
526  }
527  }
528  }
529  }
530  }
531  }
532  }
533  }
534  }
535  }
536  }
537  }
538  }
539  }
540  }
541  }
542  }
543  }
544  }
545  }
546  }
547  }
548  }
549  }
550  }
551  }
552  }
553  }
554  }
555  }
556  }
557  }
558  }
559  }
560  }
561  }
562  }
563  }
564  }
565  }
566  }
567  }
568  }
569  }
570  }
571  }
572  }
573  }
574  }
575  }
576  }
577  }
578  }
579  }
580  }
581  }
582  }
583  }
584  }
585  }
586  }
587  }
588  }
589  }
590  }
591  }
592  }
593  }
594  }
595  }
596  }
597  }
598  }
599  }
600  }
601  }
602  }
603  }
604  }
605  }
606  }
607  }
608  }
609  }
610  }
611  }
612  }
613  }
614  }
615  }
616  }
617  }
618  }
619  }
620  }
621  }
622  }
623  }
624  }
625  }
626  }
627  }
628  }
629  }
630  }
631  }
632  }
633  }
634  }
635  }
636  }
637  }
638  }
639  }
640  }
641  }
642  }
643  }
644  }
645  }
646  }
647  }
648  }
649  }
650  }
651  }
652  }
653  }
654  }
655  }
656  }
657  }
658  }
659  }
660  }
661  }
662  }
663  }
664  }
665  }
666  }
667  }
668  }
669  }
670  }
671  }
672  }
673  }
674  }
675  }
676  }
677  }
678  }
679  }
680  }
681  }
682  }
683  }
684  }
685  }
686  }
687  }
688  }
689  }
690  }
691  }
692  }
693  }
694  }
695  }
696  }
697  }
698  }
699  }
700  }
701  }
702  }
703  }
704  }
705  }
706  }
707  }
708  }
709  }
710  }
711  }
712  }
713  }
714  }
715  }
716  }
717  }
718  }
719  }
720  }
721  }
722  }
723  }
724  }
725  }
726  }
727  }
728  }
729  }
730  }
731  }
732  }
733  }
734  }
735  }
736  }
737  }
738  }
739  }
740  }
741  }
742  }
743  }
744  }
745  }
746  }
747  }
748  }
749  }
750  }
751  }
752  }
753  }
754  }
755  }
756  }
757  }
758  }
759  }
760  }
761  }
762  }
763  }
764  }
765  }
766  }
767  }
768  }
769  }
770  }
771  }
772  }
773  }
774  }
775  }
776  }
777  }
778  }
779  }
780  }
781  }
782  }
783  }
784  }
785  }
786  }
787  }
788  }
789  }
790  }
791  }
792  }
793  }
794  }
795  }
796  }
797  }
798  }
799  }
800  }
801  }
802  }
803  }
804  }
805  }
806  }
807  }
808  }
809  }
810  }
811  }
812  }
813  }
814  }
815  }
816  }
817  }
818  }
819  }
820  }
821  }
822  }
823  }
824  }
825  }
826  }
827  }
828  }
829  }
830  }
831  }
832  }
833  }
834  }
835  }
836  }
837  }
838  }
839  }
840  }
841  }
842  }
843  }
844  }
845  }
846  }
847  }
848  }
849  }
850  }
851  }
852  }
853  }
854  }
855  }
856  }
857  }
858  }
859  }
860  }
861  }
862  }
863  }
864  }
865  }
866  }
867  }
868  }
869  }
870  }
871  }
872  }
873  }
874  }
875  }
876  }
877  }
878  }
879  }
880  }
881  }
882  }
883  }
884  }
885  }
886  }
887  }
888  }
889  }
890  }
891  }
892  }
893  }
894  }
895  }
896  }
897  }
898  }
899  }
900  }
901  }
902  }
903  }
904  }
905  }
906  }
907  }
908  }
909  }
910  }
911  }
912  }
913  }
914  }
915  }
916  }
917  }
918  }
919  }
920  }
921  }
922  }
923  }
924  }
925  }
926  }
927  }
928  }
929  }
930  }
931  }
932  }
933  }
934  }
935  }
936  }
937  }
938  }
939  }
940  }
941  }
942  }
943  }
944  }
945  }
946  }
947  }
948  }
949  }
950  }
951  }
952  }
953  }
954  }
955  }
956  }
957  }
958  }
959  }
960  }
961  }
962  }
963  }
964  }
965  }
966  }
967  }
968  }
969  }
970  }
971  }
972  }
973  }
974  }
975  }
976  }
977  }
978  }
979  }
980  }
981  }
982  }
983  }
984  }
985  }
986  }
987  }
988  }
989  }
990  }
991  }
992  }
993  }
994  }
995  }
996  }
997  }
998  }
999  }
1000 }
```

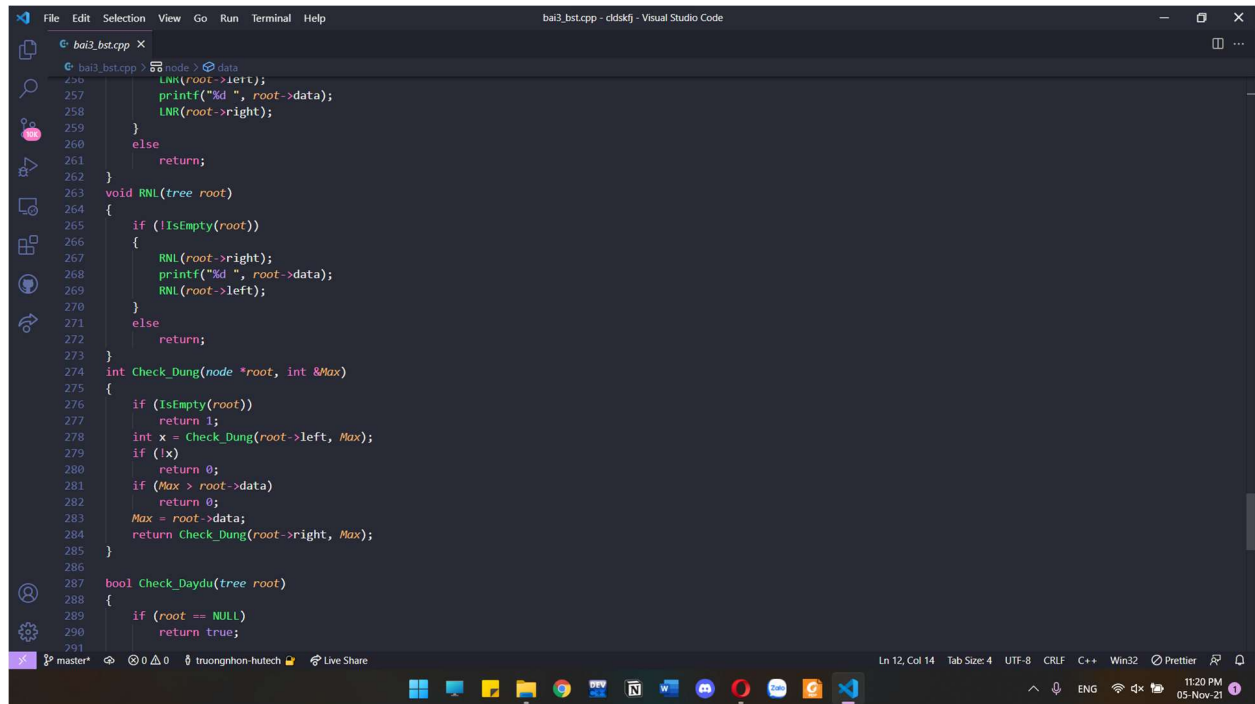
```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code
bai3_bst.cpp
163 int IsEmpty(tree root)
164 {
165     return root == NULL;
166 }
167 tree CreateNode(int x)
168 {
169     tree p = new node;
170     p->data = x;
171     p->left = p->right = NULL;
172     return p;
173 }
174 void InsertNode(tree &root, int x)
175 {
176     if (IsEmpty(root))
177     {
178         root = CreateNode(x);
179         return;
180     }
181     else
182     {
183         if (root->data == x)
184             return;
185         else if (root->data > x)
186             InsertNode(root->left, x);
187         else
188             InsertNode(root->right, x);
189     }
190 }
191 void CreateTree(tree &root)
192 {
193     int x, n;
194     printf("Nhap so luong node: ");
195     scanf("%d", &n);
196     for (int i = 0; i < n; i++)
197     {
198         // ...
199     }
200 }
201
202 int Remove(tree &root, int x)
203 {
204     if (root == NULL)
205         return 0;
206     if (root->data > x)
207         return Remove(root->left, x);
208     if (root->data < x)
209         return Remove(root->right, x);
210     node *p, *f, *rp;
211     p = root;
212     if (root->left == NULL)
213     {
214         root = root->right;
215         delete p;
216     }
217     else if (root->right == NULL)
218     {
219         root = root->left;
220         delete p;
221     }
222     else
223     {
224         f = p;
225         rp = p->right;
226         while (rp->left != NULL)
227             rp = rp->left;
228         rp->left = p;
229         delete p;
230     }
231 }
```

```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code
bai3_bst.cpp
195 scanf("%d", &n);
196 for (int i = 0; i < n; i++)
197 {
198     printf("\nNhap gia tri cho node %d: ", i);
199     scanf("%d", &x);
200     InsertNode(root, x);
201 }
202
203 int Remove(tree &root, int x)
204 {
205     if (root == NULL)
206         return 0;
207     if (root->data > x)
208         return Remove(root->left, x);
209     if (root->data < x)
210         return Remove(root->right, x);
211     node *p, *f, *rp;
212     p = root;
213     if (root->left == NULL)
214     {
215         root = root->right;
216         delete p;
217     }
218     else if (root->right == NULL)
219     {
220         root = root->left;
221         delete p;
222     }
223     else
224     {
225         f = p;
226         rp = p->right;
227         while (rp->left != NULL)
228             rp = rp->left;
229         rp->left = p;
230         delete p;
231     }
232 }
```



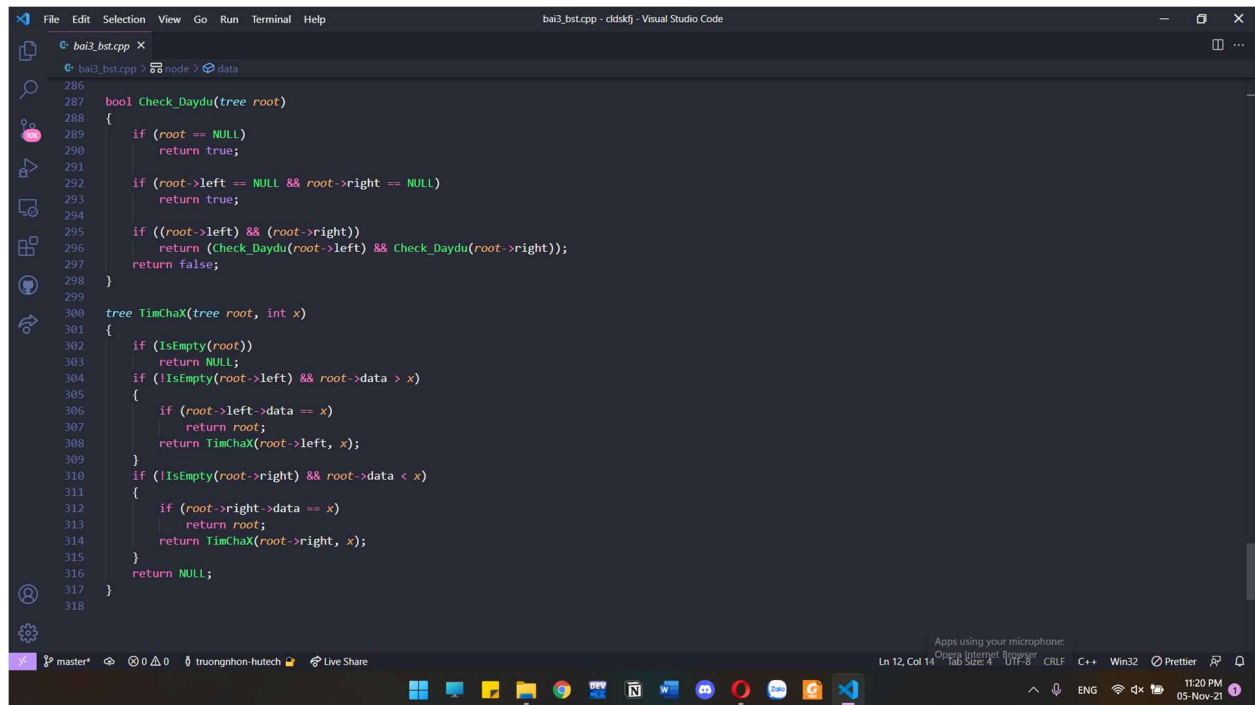
```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code

G bai3_bst.cpp X
G bai3_bst.cpp > node > data
223
224     else
225     {
226         f = p;
227         rp = p->right;
228         while (rp->left != NULL)
229         {
230             f = rp;
231             rp = rp->left;
232         }
233         p->data = rp->data;
234         if (f == p)
235             p->right = rp->right;
236         else
237             f->left = rp->right;
238         delete rp;
239     }
240     return 1;
241 }
242
243 void cleartree(tree root)
244 {
245     if (root != NULL)
246     {
247         cleartree(root->left);
248         cleartree(root->right);
249         delete root;
250     }
251 }
252 void LNR(tree root)
253 {
254     if (!IsEmpty(root))
255     {
256         LNR(root->left);
257         printf("%d ", root->data);
258         LNR(root->right);
259     }
260     else
261         return;
262 }
263 void RNL(tree root)
264 {
265     if (!IsEmpty(root))
266     {
267         RNL(root->right);
268         printf("%d ", root->data);
269         RNL(root->left);
270     }
271     else
272         return;
273 }
274 int Check_Dung(node *root, int &Max)
275 {
276     if (IsEmpty(root))
277         return 1;
278     int x = Check_Dung(root->left, Max);
279     if (!x)
280         return 0;
281     if (Max > root->data)
282         return 0;
283     Max = root->data;
284     return Check_Dung(root->right, Max);
285 }
286
287 bool Check_Daydu(tree root)
288 {
289     if (root == NULL)
290         return true;
291 }
```



```
File Edit Selection View Go Run Terminal Help
bai3_bst.cpp - cldskjf - Visual Studio Code

G bai3_bst.cpp X
G bai3_bst.cpp > node > data
256     LNR(root->left);
257     printf("%d ", root->data);
258     LNR(root->right);
259 }
260 else
261     return;
262 }
263 void RNL(tree root)
264 {
265     if (!IsEmpty(root))
266     {
267         RNL(root->right);
268         printf("%d ", root->data);
269         RNL(root->left);
270     }
271     else
272         return;
273 }
274 int Check_Dung(node *root, int &Max)
275 {
276     if (IsEmpty(root))
277         return 1;
278     int x = Check_Dung(root->left, Max);
279     if (!x)
280         return 0;
281     if (Max > root->data)
282         return 0;
283     Max = root->data;
284     return Check_Dung(root->right, Max);
285 }
286
287 bool Check_Daydu(tree root)
288 {
289     if (root == NULL)
290         return true;
291 }
```



```
286
287 bool Check_Daydu(tree root)
288 {
289     if (root == NULL)
290         return true;
291
292     if (root->left == NULL && root->right == NULL)
293         return true;
294
295     if ((root->left && (root->right))
296         return (Check_Daydu(root->left) && Check_Daydu(root->right));
297     return false;
298 }
299
300 tree TimChax(tree root, int x)
301 {
302     if (IsEmpty(root))
303         return NULL;
304     if (IsEmpty(root->left) && root->data > x)
305     {
306         if (root->left->data == x)
307             return root;
308         return TimChax(root->left, x);
309     }
310     if (IsEmpty(root->right) && root->data < x)
311     {
312         if (root->right->data == x)
313             return root;
314         return TimChax(root->right, x);
315     }
316     return NULL;
317 }
318
```

Chủ đề 4: CÂY NHỊ PHÂN TÌM KIẾM CÂN BẰNG

Câu 1: Cài đặt cấu trúc dữ liệu liên kết cho cây AVL, với các thao tác:

- a) Cài đặt các thao tác xây dựng cây: Init, IsEmpty, CreateNode
- b) Cài đặt thao tác cập nhật: Insert.

Bài làm:

```
File Edit Selection View Go Run Terminal Help
Bai4_avl.cpp - cldiskj - Visual Studio Code

C: Bai4_avl.cpp X
C: Bai4_avl.cpp > ...
1 // Phạm Đức Tài
2 // 2080601324
3 // 08/03/2002
4 // taidotcpp@gmail.com
5
6 #include <stdio.h>
7
8 struct Node
9 {
10     int data, h;
11     Node *l, *r;
12 };
13 void Init(Node *x)
14 {
15     x = NULL;
16 }
17 int max(int x, int y)
18 {
19     return x > y ? x : y;
20 }
21 int IsEmpty(Node *x)
22 {
23     return x == NULL;
24 }
25 Node *CreateNode(int data)
26 {
27     Node *q = new Node;
28     q->data = data;
29     q->h = 1;
30     Init(q->l);
31     Init(q->r);
32     return q;
33 }
34 int GetHeight(Node *x)
35 {
36     if (IsEmpty(x))
37         return 0;
38     return max(GetHeight(x->l), GetHeight(x->r)) + 1;
39 }
```

```
File Edit Selection View Go Run Terminal Help
Bai4_avl.cpp - cldiskj - Visual Studio Code

C: Bai4_avl.cpp X
C: Bai4_avl.cpp > Init(Node *x)
34 {
35     if (IsEmpty(x))
36         return 0;
37     return max(GetHeight(x->l), GetHeight(x->r)) + 1;
38 }
39 int GetBalance(Node *x)
40 {
41     return GetHeight(x->l) - GetHeight(x->r);
42 }
43
44 void LeftRotate(Node *x)
45 {
46     Node *y = x->r;
47     Node *p = y->l;
48     x->r = p;
49     y->l = x;
50     x->h = max(GetHeight(x->l), GetHeight(x->r)) + 1;
51     y->h = max(GetHeight(y->l), GetHeight(y->r)) + 1;
52     x = y;
53 }
54
55 void RightRotate(Node *x)
56 {
57     Node *y = x->l;
58     Node *p = y->r;
59     x->l = p;
60     y->r = x;
61     x->h = max(GetHeight(x->l), GetHeight(x->r)) + 1;
62     y->h = max(GetHeight(y->l), GetHeight(y->r)) + 1;
63     x = y;
64 }
65 void Insert(Node *x, int data)
66 {
67     if (IsEmpty(x))
68     {
69         x = CreateNode(data);
70         Init(x);
71     }
72     else
73     {
74         if (GetBalance(x) > 1)
75             LeftRotate(x);
76         else if (GetBalance(x) < -1)
77             RightRotate(x);
78         Insert(x->l, data);
79         Insert(x->r, data);
80     }
81 }
```

```
File Edit Selection View Go Run Terminal Help
Bai4_avl.cpp - cldskjf - Visual Studio Code

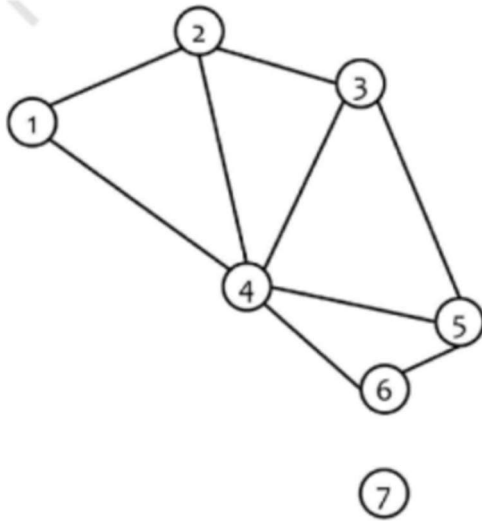
C: Bai4_avl.cpp > init(Node *B)
void Insert(Node *&x, int data)
{
    if (IsEmpty(x))
    {
        x = CreateNode(data);
        return;
    }
    if (data < x->data)
        Insert(x->l, data);
    else if (data > x->data)
        Insert(x->r, data);
    else
        return;
    int B = GetBalance(x);
    if (B > 1 && data < x->l->data)
    {
        RightRotate(x);
        return;
    }
    if (B > 1 && data > x->l->data)
    {
        LeftRotate(x->l);
        RightRotate(x);
        return;
    }
    if (B < -1 && data > x->r->data)
    {
        LeftRotate(x);
        return;
    }
    if (B < -1 && data < x->r->data)
    {
        RightRotate(x->r);
        LeftRotate(x);
        return;
    }
}
```

```
File Edit Selection View Go Run Terminal Help
Bai4_avl.cpp - cldskjf - Visual Studio Code

C: Bai4_avl.cpp > init(Node *B)
return;
}
x->h = max(GetHeight(x->l), GetHeight(x->r)) + 1;
}
void NLR(Node *x)
{
    if (IsEmpty(x))
        return;
    printf("%d ", x->data);
    NLR(x->l);
    NLR(x->r);
}
int main()
{
    Node *root;
    Init(root);
    int n;
    printf("Nhap n: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        int data;
        printf("Nhap so thu %d: ", i + 1);
        scanf("%d", &data);
        Insert(root, data);
        NLR(root);
        printf("\n");
    }
    printf("\nKet qua cuoi cung: ");
    NLR(root);
    return 0;
}
```

Chủ đề 5: ĐỒ THỊ - BIỂU DIỄN ĐỒ THỊ

Câu 1: Hãy biểu diễn đồ thị sau sử dụng ma trận kề, danh sách cạnh (cung), và danh sách kề.



Những cặp cạnh của đồ thị: $(1, 2)$, $(2, 3)$, $(3, 5)$, $(5, 6)$, $(1, 4)$, $(2, 4)$, $(3, 4)$, $(4, 5)$, $(4, 6)$.

- Ma trận kề:

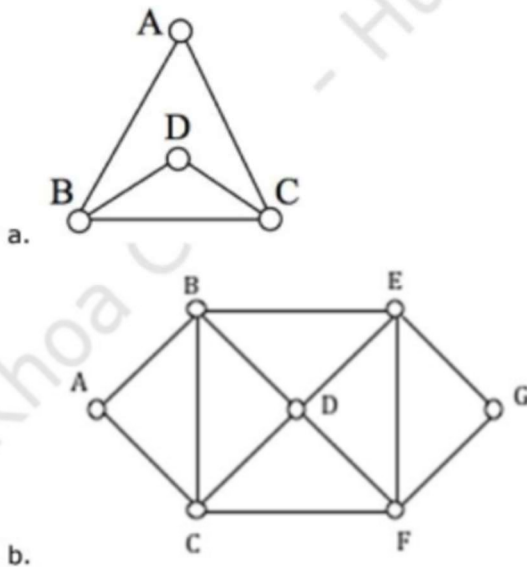
0	1	2	3	4	5	6	7
1	0	1	0	1	0	0	0
2	1	0	1	1	0	0	0
3	0	1	0	1	1	0	0
4	1	1	1	0	1	1	0
5	0	0	1	1	0	1	0
6	0	0	0	1	1	0	0
7	0	0	0	0	0	0	0

- Danh sách kề:

- 1 -> 2 -> 4
- 2 -> 1 -> 3 -> 4
- 3 -> 2 -> 4 -> 5
- 4 -> 1 -> 2 -> 3 -> 5
- 5 -> 3 -> 4 -> 6
- 6 -> 4 -> 5
- 7

Chủ đề 6: ĐƯỜNG ĐI VÀ CHU TRÌNH

Câu 1: Sử dụng thuật toán Fleury xác định chu trình Euler (nếu có) của các đồ thị sau:



a)

- Đồ thị có 2 đỉnh bậc lẻ : B, C và đồ thị liên thông

-> đồ thị có đường đi euler.

- Đường đi euler: B -> A -> C -> D -> B -> C

b)

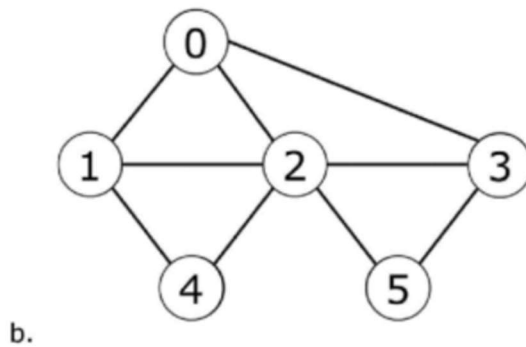
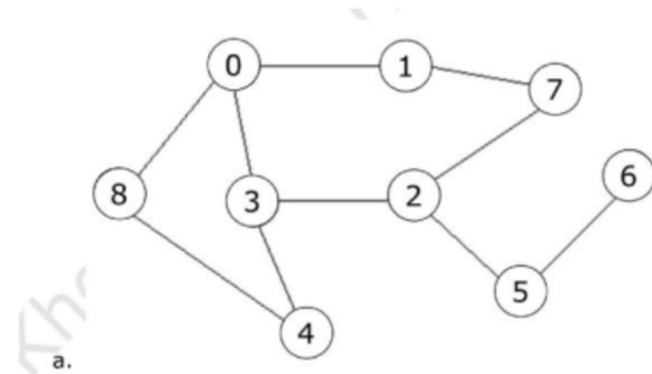
- Đồ thị b có tất cả các đỉnh là bậc chẵn và đồ thị liên thông

-> đồ thị có chu trình euler.

- Chu trình euler: A->B->C->D->B->E->D->F->E->G->F->C->A

Chủ đề 7: DUYỆT ĐỒ THỊ

Câu 1: Áp dụng giải thuật duyệt cây theo chiều rộng (BFS) với các đồ thị sau. Đỉnh xuất phát là đỉnh 0 .



a)

queue	v	w	V	queue sau khi thêm
0	0	1 8	0 1 8	1 8
1 8	1	7	0 1 8 7	8 7
8 7	8	4	0 1 8 7 4	7 4
7 4	7	2	0 1 8 7 4 2	4 2
4 2	4	3	0 1 8 7 4 2 3	2 3
2 3	2	5	0 1 8 7 4 2 3 5	3 5
3 5	3	rỗng	0 1 8 7 4 2 3 5	5
5	5	6	0 1 8 7 4 2 3 5 6	6
6	6	Rỗng	0 1 8 7 4 2 3 5 6	Hết

v: đỉnh đang xét
w: đỉnh kề
V: tập đỉnh đi qua

Vậy thứ tự duyệt cây theo chiều rộng(BFS) là: 0 -> 1-> 8 -> 7 -> 4 -> 2 -> 3 -> 5.

b)

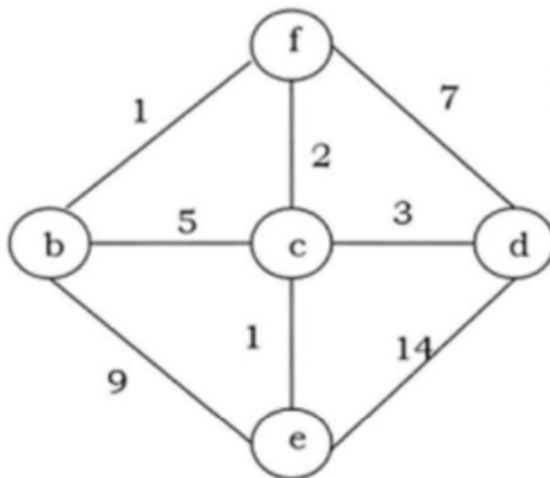
queue	v	w	V	queue sau khi thêm
0	0	1 2 3	0 1 2 3	1 2 3
1 2 3	1	4	0 1 2 3 4	2 3 4
2 3 4	2	5	0 1 2 3 4 5	3 4 5
3 4 5	3	Rỗng	0 1 2 3 4 5	4 5
4 5	4	rỗng	0 1 2 3 4 5	5
5	5	rỗng	0 1 2 3 4 5	rỗng

v: đỉnh đang xét
w: đỉnh kề
V: tập đỉnh đi qua

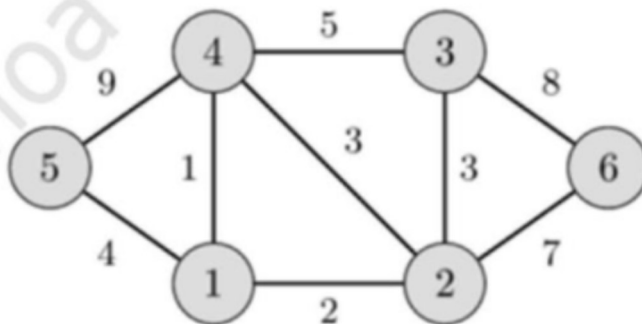
Vậy thứ tự duyệt cây theo chiều rộng(BFS) là: 0 -> 1 -> 2 -> 3 -> 4 -> 5.

Chủ đề 8: CÂY BAO TRÙM VÀ CÂY BAO TRÙM NHỎ NHẤT

Câu 1. Tìm cây bao trùm nhỏ nhất của các đồ thị sau bằng thuật toán Kruskal



a.



b.

a) Đồ thị a:

sắp xếp	T
[b, f](1)	[b, f](1)
[c, e](1)	[c, e](1)
[c, f](2)	[c, f](2)
[c, d](3)	[c, d](3)
[c, b](5)	
[d, f](7)	
[b, e](9)	
[e, d](14)	

$$\text{Tổng trọng số} = 1 + 1 + 2 + 3 = 7$$

Vậy cây khung nhỏ nhất của đồ thị a là: [b, f], [c, e], [c, f], [c, d].

b) Đồ thị b:

sắp xếp	T
[1, 4](1)	[1, 4](1)
[1, 2](2)	[1, 2](2)
[2, 4](3)	[2, 3](3)
[2, 3](3)	[1, 5](4)
[1, 5](4)	[2, 6](7)
[3, 4](5)	
[2, 6](7)	
[3, 6](8)	
[4, 5](9)	

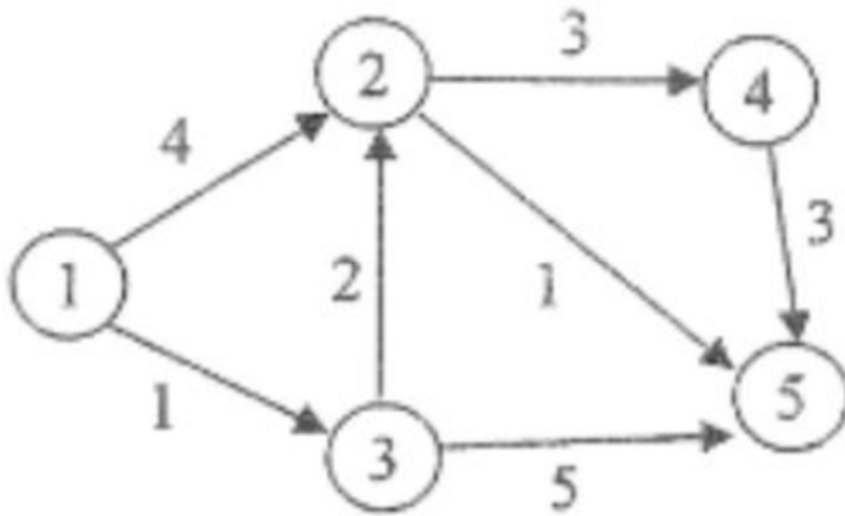
$$\text{Tổng trọng số} = 1 + 2 + 3 + 4 + 7 = 17$$

Vậy
3], [1, 5], [2, 6].

cây khung nhỏ nhất của đồ thị b là: [1, 4], [1, 2], [2,

Chủ đề 9: ĐƯỜNG ĐI NGẮN NHẤT

Câu 1. Áp dụng giải thuật Dijkstra tìm đường đi ngắn nhất từ đỉnh có chỉ số nhỏ nhất tới các đỉnh còn lại của các đồ thị sau:



Chọn 1 làm gốc ta lập bảng:

	1	2	3	4	5
khởi tạo	1*	(vc,-)	(vc,-)	(vc,-)	(vc,-)
1	-	(4,1)	(1,1)	(vc,-)	(vc,-)
2	-	(2,3)	-	(vc,-)	(5,3)
3	-	-	-	(3,2)	(1,2)
4	-	-	-	(3,2)	-
5	-	-	-	-	-

Đường đi ngắn nhất từ 1 đến các đỉnh:

1 -> 3 (1)

1 -> 3 -> 2 (3)

1 -> 3 -> 2 -> 5 (4)

1 -> 3 -> 2 -> 4 (6)