

Duyệt đồ thị theo chiều sâu

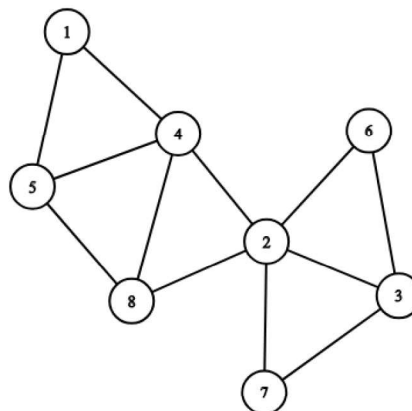
```
1  /* Khai bao Stack*/
2  #define MAX_ELEMENTS 100
3  typedef struct {
4      int data[MAX_ELEMENTS];
5      int size;
6  } Stack;
7
8  void make_null_stack(Stack* S) {
9      S->size = 0;
10 }
11 void push(Stack* S, int x) {
12     S->data[S->size] = x;
13     S->size++;
14 }
15 int top(Stack* S) {
16     return S->data[S->size - 1];
17 }
18 void pop(Stack* S) {
19     S->size--;
20 }
21 int empty(Stack* S) {
22     return S->size == 0;
23 }
```

```

1  /* Duyệt đồ thị theo chiều sâu */
2  void depth_first_search(Graph* G) {
3      Stack frontier;
4      int mark[MAX_VERTEXES];
5      make_null_stack(&frontier);
6      /* Khởi tạo mark, chưa đỉnh nào được xét */
7      int j;
8      for (j = 1; j <= G->n; j++)
9          mark[j] = 0;
10     /* Đưa 1 vào frontier */
11     push(&frontier, 1);
12     /* Vòng lặp chính dùng để duyệt */
13     while (!empty(&frontier)) {
14         /* Lấy phần tử đầu tiên trong frontier ra */
15         int x = top(&frontier); pop(&frontier);
16         if (mark[x]!=0) //Đã được duyệt rồi
17             continue;
18         printf("Duyệt %d\n", x);
19         mark[x]=1; //Đánh dấu đã được duyệt rồi
20         /* Lấy các đỉnh kề của nó */
21         List list = neighbors(G, x);
22         /* Xét các đỉnh kề của nó */
23         for (j = 1; j <= list.size; j++) {
24             int y = element_at(&list, j);
25             push(&frontier, y);
26         }
27     }
28 }
29

```

Bài tập: Chạy từng dòng code bằng tay và thực hiện duyệt đồ thị sau đây theo chiều sâu:



Duyệt đồ thị theo chiều rộng

```
1  /* Khai bao Queue */
2  #define MAX_ELEMENTS 100
3  typedef struct {
4      int data[MAX_ELEMENTS];
5      int front, rear;
6  } Queue;
7  void make_null_queue(Queue* Q) {
8      Q->front = 0;
9      Q->rear = -1;
10 }
11 void push(Queue* Q, int x) {
12     Q->rear++;
13     Q->data[Q->rear] = x;
14 }
15 int top(Queue* Q) {
16     return Q->data[Q->front];
17 }
18 void pop(Queue* Q) {
19     Q->front++;
20 }
21 int empty(Queue* Q) {
22     return Q->front > Q->rear;
23 }
```

```

1  /* Duyệt đồ thị theo chiều rộng */
2  void breath_first_search(Graph* G) {
3      Queue frontier;
4      int mark[MAX_VERTEXES];
5      make_null_queue(&frontier);
6      /* Khởi tạo mark, chưa đỉnh nào được xét */
7      int j;
8      for (j = 1; j <= G->n; j++)
9          mark[j] = 0;
10     /* Đưa 1 vào frontier */
11     push(&frontier, 1);
12     mark[1] = 1;
13     /* Vòng lặp chính dùng để duyệt */
14     while (!empty(&frontier)) {
15         /* Lấy phần tử đầu tiên trong frontier ra */
16         int x = top(&frontier); pop(&frontier);
17         printf("Duyet %d\n", x);
18         /* Lấy các đỉnh kề của nó */
19         List list = neighbors(G, x);
20         /* Xét các đỉnh kề của nó */
21         for (j = 1; j <= list.size; j++) {
22             int y = element_at(&list, j);
23             if (mark[y] == 0) {
24                 mark[y] = 1;
25                 push(&frontier, y);
26             }
27         }
28     }
29 }

```

Bài tập: Chạy từng dòng code bằng tay và thực hiện duyệt đồ thị sau đây theo chiều rộng:

