

[Nhà của tôi](#) / [Các khoá học của tôi](#) / [LTDI HK1 2526 \(GV PXHIEN\)](#) / [Thực hành](#) / [Buổi 1](#)

Câu hỏi **7**

Đúng

Đạt điểm 1,00

Cho cấu trúc dữ liệu đồ thị được khai báo sử dụng ma trận đỉnh - cung như sau:

```
typedef struct {
    int A[100][500];
    int n, m;
} Graph;
```

Giả sử đồ thị **không chứa khuyên**.

Viết hàm **List neighbors(Graph* G, int x)** trả về danh sách các đỉnh kề của x.

Chú ý: các đỉnh kề của x được sắp xếp theo thứ tự tăng dần và không trùng nhau. Ví dụ: nếu các đỉnh kề của 1 là 4 và 2 thì danh sách trả về chứa: 2 và 4.

Cấu trúc dữ liệu List được định nghĩa như bên dưới:

```
#define MAX_ELEMENTS 100
typedef int ElementType;
typedef struct {
    ElementType data[MAX_ELEMENTS];
    int size;
} List;
/* Tạo danh sách rỗng */
void make_null(List* L) {
    L->size = 0;
}
/* Thêm một phần tử vào cuối danh sách */
void push_back(List* L, ElementType x) {
    L->data[L->size] = x;
    L->size++;
}
/* Lấy phần tử tại vị trí i, phần tử bắt đầu ở vị trí 1 */
ElementType element_at(List* L, int i) {
    return L->data[i-1];
}
/* Trả về số phần tử của danh sách */
int count_list(List* L) {
    return L->size;
}
```

For example:

Input	Result
4 5	2
1 2	3
1 3	4
1 4	
2 3	
1 3	

Answer: (penalty regime: 33.3, 66.7, ... %)

```
1 List neighbors(Graph* G, int x) {
2     List L;
3     make_null(&L);
4
5     for (int e = 1; e <= G->m; e++) {
6         if (G->A[x][e] == 1) {
7             // Tìm đỉnh còn lại trong cùng e
8             for (int y = 1; y <= G->n; y++) {
9                 if (y != x && G->A[y][e] == 1) {
```

```

10         // kiểm tra tránh trùng
11         int found = 0;
12         for (int k = 1; k <= count_list(&L); k++) {
13             if (element_at(&L, k) == y) {
14                 found = 1;
15                 break;
16             }
17         }
18         if (!found) push_back(&L, y);
19     }
20 }
21 }
22 }
23
24 // Sắp xếp tăng dần danh sách L (Bubble sort đơn giản)
25 for (int i = 0; i < L.size - 1; i++) {
26     for (int j = i + 1; j < L.size; j++) {
27         if (L.data[i] > L.data[j]) {
28             int tmp = L.data[i];
29             L.data[i] = L.data[j];
30             L.data[j] = tmp;
31         }
32     }
33 }
34
35 return L;
36 }
37

```

	Input	Expected	Got	
✓	4 5	2	2	✓
	1 2	3	3	
	1 3	4	4	
	1 4			
	2 3			
	1 3			

Passed all tests! ✓

◀ tt

Chuyển tới...

Buổi 2 - phần cơ bản ▶