

Các đề bài tập lập trình môn Mạng máy tính

Trong nội dung môn học, sinh viên sẽ được giới thiệu các thuật toán sử dụng trong mạng máy tính, sinh viên chọn để lập trình thể hiện các thuật toán này.

1. Thuật toán Dijkstra tìm đường đi ngắn nhất từ 1 đỉnh đến các đỉnh khác trong đồ thị có trọng số.

Yêu cầu:

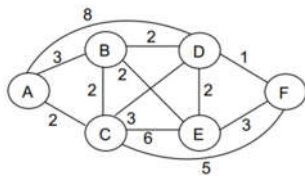
Input: Các trọng số của đồ thị

Output:

a. Bảng kết quả của thuật toán thể hiện khoảng cách ngắn nhất từ đỉnh xuất phát đến tất cả các đỉnh còn lại.

b. In đường đi ngắn nhất, ví dụ 1->3->4->5 có giá là 6

Ví dụ: Cho đồ thị



Kết quả (output)

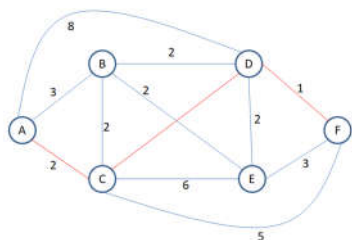
Bước	Tập N	2(B)	3(C)	4(D)	5(E)	6(F)
0	1	[3,1]	*[2,1]	[8,1]	∞	∞
1	1,3	*[3,1]		[5,3]	[8,3]	[7,3]
2	1,3,2			*[5,3]	[5,2]	[7,3]
3	1,3,2,4				*[5,2]	[6,4]
4	1,3,2,4,5					*[6,4]
5	1,3,2,4,5,6					

Độ dài ngắn nhất của đường đi từ 1(A) đến 6(F) là 3 theo thứ tự 1(A)->3(C)->4(D)->6(F).

Lưu ý: Vì thuật toán này đã có nhiều code trên Internet để tham khảo nên sinh viên phải có cải tiến mới được điểm, một số cải tiến ví dụ như:

1) Lập trình đồ họa (tối đa 4 điểm): cho phép người dùng chọn số đỉnh của đồ thị (tối đa 10 đỉnh) sau đó sẽ vẽ các đỉnh ngẫu nhiên trên màn hình, người dùng có thể chọn các cạnh nối giữa các đỉnh và trọng số ghi trên cạnh tương ứng hoặc để chương trình tự sinh các cạnh này.

Sau khi thuật toán kết thúc, đường đi ngắn nhất sẽ được vẽ trên đồ thị bằng đường khác màu, ví dụ.



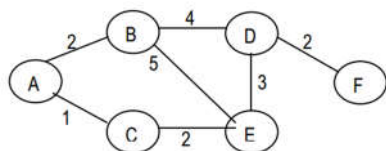
2) Thuật toán cho phép hiển thị bảng kết quả thuật toán với tất cả các đỉnh xuất phát khác nhau, ví dụ trên đã có bảng kết quả cho đỉnh xuất phát là A, tiếp tục cho các đỉnh B, C, D, E, F. (tối đa 1 điểm)

2. Thuật toán Bellman-Ford tìm đường đi ngắn nhất từ 1 đỉnh đến các đỉnh khác trong đồ thị có trọng số.

Input: Các trọng số của đồ thị

Output: Bảng kết quả của thuật toán thể hiện khoảng cách ngắn nhất từ đỉnh xuất phát đến tất cả các đỉnh còn lại.

Ví dụ: Cho đồ thị



Kết quả thuật toán với đỉnh xuất phát là A

$D^A()$	B	C
B	2	4
C	5	1
D	6	6
E	7	3
F	8	8

Lưu ý:

Mã nguồn cho thuật toán này đã có nhiều trên Internet. Chương trình sẽ phải in ra bảng kết quả của thuật toán cho tất cả các đỉnh (1 điểm)

Yêu cầu nâng cao như thuật toán Dijkstra ở trên (lập trình đồ họa tối đa 4 điểm).

3. Thuật toán tính UDP checksum 16 bit với các từ dữ liệu.

Ví dụ tính UDP Checksum

☐ Note

- Khi cộng hai số, nhớ ở bit cao nhất sẽ được cộng vào kết quả
- Ví dụ: Cộng hai số nguyên 16-bit

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Dữ	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
Tổng	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0
Tổng kiểm tra	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Input: Từ tiếng Anh cần tính checksum

Output: Checksum tính được

Ví dụ:

Nhập dữ liệu cần gửi đi: **THE**

Checksum: **0110011010110111**

4. Thuật toán Hamming:

4.1 Tìm mã Hamming của 1 ký tự 8 bit (1 điểm)

Input: Ký tự chữ cái tiếng Anh (A-z)

Output: Mã Hamming của ký tự nhập

Ví dụ: Nhập ký tự: A

Mã hamming của A là: 100010010001

4.2 Tìm 1 bit lỗi trong mã Hamming của 1 ký tự, sửa lỗi và khôi phục mã nhị phân của ký tự ban đầu và xác định ký tự ban đầu (1,5 điểm).

Input: Mã Hamming của một ký tự có 1 bit lỗi

Output: Mã Hamming sau khi đã sửa lỗi và ký tự ban đầu (nếu là ký tự hiển thị được).

Ví dụ: Nhập mã Hamming có lỗi 1 bit: 110010100010

Mã nhị phân ban đầu là: 01010110, ký tự ban đầu là V

5. Thuật toán CRC (1,5 điểm)

Input: Từ tiếng Anh cần truyền (ví dụ TEST) và chuỗi bit G (ví dụ 10110).

Output: Chuỗi bit R theo thuật toán CRC.

Ví dụ:

Nhập dữ liệu cần truyền: BEST

Nhập đa thức sinh G: 10101

R theo giải thuật CRC là: 1001

6. Lập trình giải các bài tập chia mạng con IPv4 (2-4 điểm), tham khảo các chương trình mẫu:

<https://www.calculator.net/ip-subnet-calculator.html?cclass=any&csubnet=30&cip=14.162.89.37&ctype=ipv4&printit=0&x=61&y=21>

<http://taimienphi.vn/author/Bitcricket>

Khuyến khích các chương trình viết theo giao diện Web (cộng thêm 1 điểm)