## Bài tập ctdlgt-bt12

### Bài 1

Làm bài [Connected Cells in a Grid](https://www.hackerrank.com/challenges/connected-cell-in-a-grid)

Tóm tắt: Cho mảng hai chiều với các phần tử (ô) có giá trị là 0 hoặc 1. Hai phần từ được gọi là kết nối (connected) với nhau nếu chúng cùng có giá trị là 1 và ở sát nhau theo chiều ngang, chiều dọc và chéo cạnh. Như thế mỗi ô có thể có nhiều nhất 8 ô ở kề nối với nó.

Các ô kết nối với nhau tạo thành vùng. Hãy tìm và in ra số phần tử của vùng lớn nhất.

Gợi ý: Xem đây là một đồ thị với đỉnh là các ô, và hai cạnh nối với nhau nếu 2 ô là kết nối (connected). Với mỗi đỉnh (ô), dùng thuật toán duyệt theo chiều sau (DFS) hoặc duyệt theo chiều rộng (BFS) để lan ra hết các đỉnh (ô) liên thông với nó. Khi duyệt, nếu đỉnh nào đã thăm rồi thì không duyệt nữa. Nói cách khác, đây là bài toán tìm thành phần liên thông của đồ thị.

Mã nguồn: Sinh viên có thể sử dụng [phần chương trình sau](https://drive.google.com/file/d/0B7mScmczCiHWRGUwWFBITmFNdWc/view?usp=sharing&resourcekey=0-9skcsa6_uslMsQ3meZHyvg);

trong đó viết sẵn việc nhập dữ liệu, khai báo mảng các ô grid[][] cũng như mảng lưu trạng thái các ô đã được thăm visited[][] chưa.

Sinh viên cần viết nốt phần hàm *static int count\_connected(int row, int col)* đếm số ô của thành phần liên thông với ô có toạ độ (row, col). Trong hàm này cũng cần cập nhật/đánh dấu các ô đã thăm để không duyệt nữa vào mảng visited[][] được khai báo ở ngoài hàm.

### Bài 2

Làm bài [Breadth First Search: Shortest Reach](https://www.hackerrank.com/challenges/bfsshortreach)

Tóm tắt: Cho đồ thị vô hướng; các đỉnh đánh số từ 1 tới n; hai cạnh nếu nối với nhau thì có trọng số luôn là 6. Cho vào đỉnh xuất phát, hãy tìm và in đường đi ngắn nhất từ đỉnh này tới (n-1) đỉnh còn lại. Nếu không có đường đi thì in ra -1.

Chú ý là dữ liệu vào có q câu hỏi, mỗi câu hỏi cho một đồ thị với một đỉnh xuất phát.

Gợi ý:

Dùng thuật toán duyệt theo chiều rộng (BFS) để tìm đường đi từ đỉnh xuất phát tới (n-1) đỉnh còn lại. (Không cần tới thuật toán Dijkstra bởi tất cả các cạnh đều có cùng trọng số là 6.)

Mã nguồn: Sinh viên có thể sử dụng [phần chương trình sau](https://drive.google.com/file/d/0B7mScmczCiHWcW8xMl9GMUxUVUE/view?usp=sharing&resourcekey=0-iMsO6K2g_Q4UbUtJ3n9Dbg);

trong đó viết sẵn việc nhập dữ liệu, khai báo và khởi tạo đồ thị. (Đồ thị được biểu diễn dạng danh sách đỉnh liền kề, với ArrayList. Các đỉnh đánh số từ 1, nên bỏ qua phần từ đầu của mảng có chỉ số là 0.)

Sinh viên cần viết nốt phần hàm *static int[] findShortestReach(ArrayList> graph, int start)* tìm đường đi ngắn nhất từ đỉnh start tới các đỉnh khác, lưu kết quả vào mảng costs[] và trả về.