

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO MÔN HỌC

Tối ưu và lập kế hoạch

Đề tài: Phân luồng cách ly COVID 19

Giảng viên hướng dẫn: Bùi Quốc Trung

Mã lớp học: 136049

Danh sách sinh viên thực hiện:

ST T	Họ và tên	MSSV	Lớp
1	Nguyễn Thành Phong	20192016	CTTN-KHMT-K64
2	Trịnh Hồng Phụng	20190062	CTTN-KHMT-K64
3	Nghiêm Việt Thắng	20190088	CTTN-KHMT-K64
4	Hữu Tường Tú	20194395	CTTN-KHMT-K64

MỤC LỤC

I, Giới thiệu đề tài	3
II, Tóm tắt bài toán	4
1, Kí hiệu.....	4
2, Biến.....	4
3, Điều kiện.....	4
4, Hàm mục tiêu	4
III, Các giải thuật tối ưu cục bộ	5
1, Local search (Heuristic)	5
1.1, Phương pháp:	5
1.2, Thực hiện:	5
2, Tabu search (MetaHeuristic).....	6
IV, Giải thuật tối ưu	6
1, Ortool	6
2, Graph.....	8
V, Dữ liệu	8
VI, Đánh giá kết quả	9
VII, Tài liệu tham khảo	11

I, Giới thiệu đề tài

Có N đoàn du khách $1, 2, 3, \dots, N$ từ các sân bay cần được đưa về các khu cách ly $1, 2, \dots, M$ trong bối cảnh COVID19. Đoàn khách i có $s(i)$ người. Khu cách ly j có thể chứa được $c(j)$ người (số chỗ là $c(j)$)

Đoàn khách thứ i nằm ở điểm i ($i=1, \dots, N$) và khu cách ly j nằm ở điểm $N+j$ ($j=1, \dots, M$)

Hãy xây dựng phương án cách ly sao cho:

1. Mỗi đoàn khách đưa về đúng 1 khu cách ly
2. Tổng số người đưa về mỗi khu cách ly không vượt quá số chỗ của khu cách ly đó.
3. Khoảng cách lớn nhất từ điểm xuất phát của đoàn khách nào đó đến khu cách ly được phân công là ngắn nhất

II, Tóm tắt bài toán

1, Kí hiệu

N : số đoàn khách

M : số khu cách ly

$S[i]$: thông tin của đoàn khách thứ i , $0 \leq i < N$

$C[i]$: thông tin của khu cách ly thứ i , $0 \leq i < M$

(các thông tin này gồm có :

location : tọa độ chiều của đối tượng

count : số lượng thành viên (với đoàn khách) hoặc số lượng chứa tối đa của khu cách ly)

Distance(a, b): Khoảng cách σ -clit từ địa điểm a đến địa điểm b

2, Biến

$X[i]$: Đoàn cách ly thứ i sẽ di chuyển tới khu cách ly $X[i]$, $0 \leq i \leq N - 1$

3, Điều kiện

$0 \leq X[i] \leq M - 1$ với mọi $0 \leq i \leq N - 1$

$X[i] \neq X[j]$ với mọi $i \neq j$, $0 \leq i, j \leq N - 1$

$C[X[i]].count \geq S[i].count$ với mọi $0 \leq i \leq N - 1$

4, Hàm mục tiêu

$$F = \text{Min}(\max(\text{Distance}(S[i].location, C[X[i]].location) \text{ với } 0 \leq i \leq N - 1))$$

Ta định nghĩa thêm maxdistance_X là vị trí i của đoàn cách ly phải di chuyển xa nhất(tức có $\text{Distance}(S[i], C[X[i]])$ lớn nhất) của 1 lời giải X.

III, Các giải thuật tối ưu cục bộ

1, Local search (Heuristic)

1.1, Phương pháp:

Khởi tạo một lời giải trong không gian tìm kiếm, từ đó đi tới các lời giải có giá trị hàm mục tiêu tốt hơn.

1.2, Thực hiện:

Lân cận của một lời giải hiện tại có rất nhiều cách định nghĩa một lân cận cho lời giải. Dựa vào bài toán đang xét nhóm định nghĩa một lân cận Y của 1 lời giải X nếu thỏa mãn 1 trong 2 điều kiện:

Điều kiện 1:

$$Y[\text{maxdistance_X}] \neq X[\text{maxdistance_X}]$$

$$Y[i] = X[i] \text{ với mọi } 0 \leq i \leq N-1, i \neq \text{maxdistance_X}$$

Điều kiện 2:

Tồn tại vị trí $\text{replace_X} \neq \text{maxdistance_X}$ và ta sẽ đổi vị trí cách ly của 2 đoàn khách này, cụ thể :

$$Y[\text{replace_X}] = X[\text{indexdistance_X}]$$

$$Y[\text{indexdistance_X}] = Y[\text{replace_X}]$$

$$Y[i] = X[i] \text{ với mọi } 0 \leq i \leq N-1, i, j \neq \text{maxdistance_X}, \text{replace_X}$$

Ví dụ:

$$N = 5, M = 8 (\text{maxindex_X} = 2)$$

I	0	1	2	3	4
X[i]	5	0	3	4	6
Y[i] (điều kiện 1)	5	0	2	4	6
Y[i] (điều kiện 2, replace_X= 1)	5	3	0	4	6

1.3 Ưu điểm:

Vấn đề trạng thái ‘shoulder’ và điểm ‘flat’ local minimize không xảy ra do hàng xóm mới không thể có hàm mục tiêu trùng với ban đầu.

Triển khai dễ dàng, tốc độ chạy nhanh.

1.4 Nhược điểm:

Dễ rơi vào cục bộ địa phương.

2, Tabu search (MetaHeuristic)

2.1. Phương pháp:

Phương pháp Tabu Search bắt đầu từ một lời giải ngẫu nhiên của bài toán, sau đó sẽ áp dụng một số các toán tử lân cận để sinh ra lời giải mới, đồng thời lưu lại các lời giải đã đi qua với mục đích thoát khỏi local-minimize, quá trình này được lặp đi lặp lại cho đến khi thỏa mãn điều kiện kết thúc của thuật toán.

2.2. Thực hiện:

Dựa vào bài toán đang xét nhóm định nghĩa một lân cận Y của 1 lời giải X nếu thỏa mãn 1 trong 2 điều kiện:

Điều kiện 1:

Tồn tại 1 vị trí replace_X sao cho

$$Y[\text{replace_X}] \neq X[\text{replace_X}]$$

$$Y[i] = X[i] \text{ với mọi } 0 \leq i \leq N-1, i \neq \text{replace_X}$$

Điều kiện 2:

Tồn tại 2 vị trí i và j và đổi vị trí cách ly tại 2 vị trí này trong X để được Y, cụ thể:

$$Y[i] = X[j]$$

$$X[j] = Y[j]$$

$$Y[k] = X[k], 0 \leq k \leq N-1, k \neq i, j$$

Việc lưu trữ các lời giải của bài toán dẫn tới việc bộ nhớ bị tăng lên một cách nhanh chóng, đồng thời việc kiểm tra hàng xóm có ở trong kho lưu trữ không cũng khó khăn. Do vậy số lượng lời giải khi lưu trữ thường giữ ở mức nhất định tùy vào độ lớn của dữ liệu đầu vào.

2.3 Ưu điểm:

Có thể thoát khỏi local-minimize

Cho kết quả thường khá tốt

2.4 Nhược điểm:

Thời gian chạy lớn, bộ nhớ sử dụng lớn

Có thể rơi vào vòng lặp khi có 2 vùng local-minimize chuyển qua lại

IV, Giải thuật tối ưu

1, Ortool

1.1 Phương pháp

Ortool được phát triển bởi Google giúp cho việc giải các bài toán CP trở nên dễ dàng. Với lớp bài toán CP, thuật toán or-tool tỏ ra hiệu trong việc giải quyết để tìm ra nghiệm tối ưu toàn cục thay vì các thuật toán heuristic, tuy vậy về mặt thời gian chạy không tính được trước dẫn tới cần thực nghiệm trên nhiều bộ dữ liệu khác nhau.

1.2 Triển khai

Với cách mô hình hóa ban đầu, hiện tại chưa tìm ra cách để cài đặt trong các 'constrain' của or-tool nên nhóm đề xuất sử dụng mảng 2 chiều để biểu diễn lời giải.

Các ràng buộc về biến của bài toán biểu diễn được dưới các mẫu 'constrain' sẵn có, tuy vậy hàm mục tiêu với hàm max không được hỗ trợ. Vì vậy nhóm đề xuất 2 phương án giải quyết.

Phương án 1:

Dựa vào hàm mục tiêu yêu cầu giảm thiểu khoảng cách di chuyển lớn nhất của một đoàn khách, do vậy ta có thể dùng phương pháp chắt nhệ phân để tìm kết quả tối ưu. Cụ thể, nếu tìm được một lời giải X đưa ra giá trị hàm mục tiêu là $f(X)$ thì những lời giải có giá trị hàm mục tiêu lớn hơn hoặc bằng X có thể bỏ qua. 'Constrain' có thể được viết lại như sau:

Biến:

$X[i][j]$: đoàn khách i di chuyển tới khu cách ly j thì $X[i][j] = 1$, ngược lại $X[i][j] = 0$ với $0 \leq i \leq N-1, 0 \leq j \leq M-1$

H : giá trị hàm mục tiêu ước lượng.

Điều kiện:

$$\sum_{i=0}^{N-1} X[i][j] \leq 1 \text{ với mọi } 0 \leq j \leq M-1$$

$$\sum_{j=0}^{M-1} X[i][j] \leq 1 \text{ với mọi } 0 \leq i \leq M-1$$

$$\sum_{j=0}^{M-1} X[i][j] * C[j].count \leq S[i].count \text{ với mọi } 0 \leq i \leq M-1$$

$$\sum_{j=0}^{M-1} X[i][j] * Distance(S[i], C[j]) \leq H \text{ với mọi } 0 \leq i \leq M-1, H$$

Hàm mục tiêu:

Tồn tại lời giải của bài toán

Phương án 2:

Với hàm max không có 'constrain' có sẵn, nhưng hoàn toàn có thể biểu diễn được dưới các constrain khác và biến mới. Ta mô hình hóa bài toán như sau:

Biến:

$X[i][j]$: đoàn khách i di chuyển tới khu cách ly j thì $X[i][j] = 1$, ngược lại $X[i][j] = 0$ với $0 \leq i \leq N-1, 0 \leq j \leq M-1$

Value[i] : biến giả kiểu int, $0 \leq i \leq N-1$

Bool_[i] : biến giả kiểu boolean, $0 \leq i \leq N-1$

Trong một lời giải X thì:

$$\text{Bool_}[i] = 1, 0 \leq i \leq N-1, i \neq \text{maxdistance_X}$$

$$\text{Value}[i] = 1, 0 \leq i \leq N-1, i \neq \text{maxdistance_X}$$

$$\text{Bool_}[\text{maxdistance_X}] = 1$$

$$\text{Value}[\text{maxdistance_X}] = \text{distance}(S[\text{maxdistance_X}], C[X[\text{maxdistance_X}]])$$

Điều kiện:

$$\sum_{i=0}^{N-1} X[i][j] \leq 1 \text{ với mọi } 0 \leq j \leq M-1$$

$$\sum_{j=0}^{M-1} X[i][j] \leq 1 \text{ với mọi } 0 \leq i \leq M-1$$

$$\sum_{j=0}^{M-1} X[i][j] * C[j].\text{count} \leq S[i].\text{count} \text{ với mọi } 0 \leq i \leq M-1$$

$$\sum_{k=0}^{M-1} X[i][k] * \text{distance}(S[i], C[k]) \geq \sum_{k=0}^{M-1} X[j][k] * \text{distance}(S[j], C[k]) \text{ với mọi } 0 \leq i, j \leq N-1 \text{ nếu như Bool_}[i] \text{ là true}$$

$$\text{Value}[i] = \sum_{k=0}^{M-1} X[i][k] * \text{distance}(S[i], C[k]) \text{ với}$$

$$\text{mọi } 0 \leq i, j \leq N-1 \text{ nếu như Bool_}[i] \text{ là true}$$

$$\text{Value}[i] = 0 \text{ với mọi } 0 \leq i, j \leq N-1 \text{ nếu như Bool_}[i] \text{ là false}$$

$$\sum_{i=0}^{N-1} \text{Bool_}[i] = 1$$

Hàm mục tiêu:

$$F = \min(\sum_{i=0}^{N-1} \text{Value}[i])$$

2, Graph

Sau khi tìm hiểu, nhóm nhận thấy rằng bài toán không nằm trong lớp bài toán NP khó. Giống như phương án 1 của thuật ortool, dựa vào phương pháp chặt nhị phân để tìm kiếm lời giải bài toán, sau đó xây dựng đồ thị 2 phía như sau với giá trị hàm mục tiêu ước lượng là H:

Tập hợp A gồm có N đỉnh đại diện cho N khu cách ly và tập hợp B có M đỉnh đại diện cho M khu cách ly.

1 Đỉnh a ở tập hợp A được nối với một đỉnh b ở tập hợp B nếu khoảng cách σ-clit từ đoàn khách a tới khu cách ly b bé hơn hoặc bằng H.

Mỗi cách ghép N đỉnh khác biệt của tập hợp A với N đỉnh khác biệt của tập hợp B tương ứng với một lời giải của bài toán với giá trị hàm mục tiêu ước lượng là H.

Sử dụng thuật toán ghép cặp cực đại trên đồ thị 2 phía để tìm số cặp ghép tối đa, nếu như có đủ N cặp ghép tức tồn tại lời giải cho giá trị H, và ngược lại.

Độ phức tạp của giải thuật : $O((N + M) * N * \log(M*N))$.

V, Dữ liệu

Dữ liệu được sinh bằng 2 cách:

- Dựa vào thuật toán được áp dụng, tạo ra các trường hợp đặc biệt dẫn tới cho kết quả sai. Nhóm dựa vào các dữ liệu này để sửa đổi cho đúng.
- Tạo ra ngẫu nhiên:

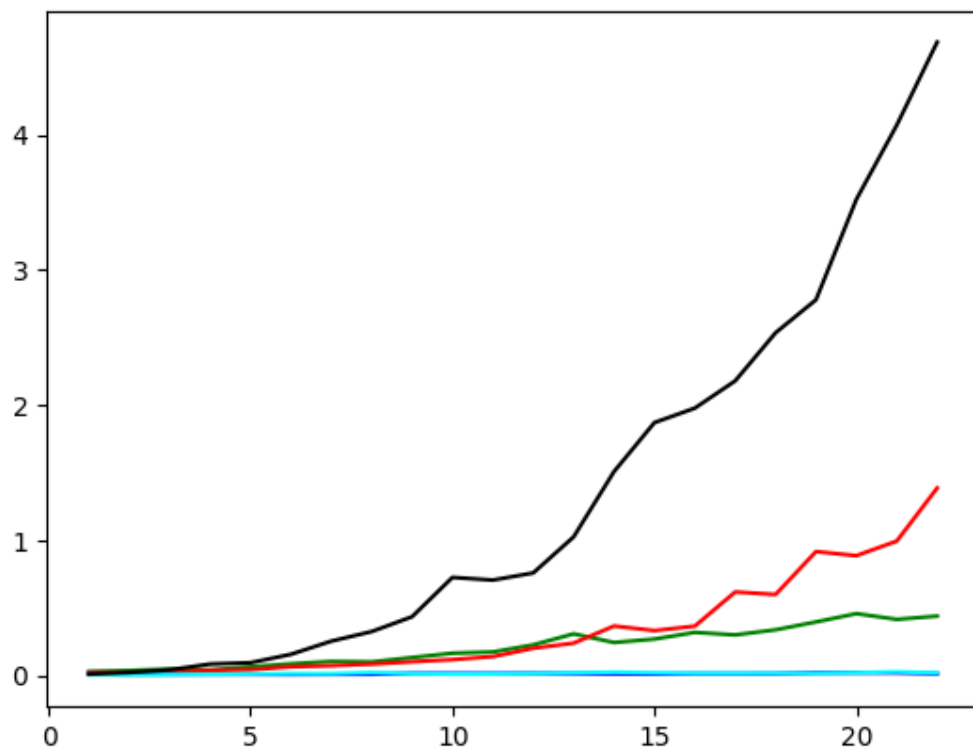
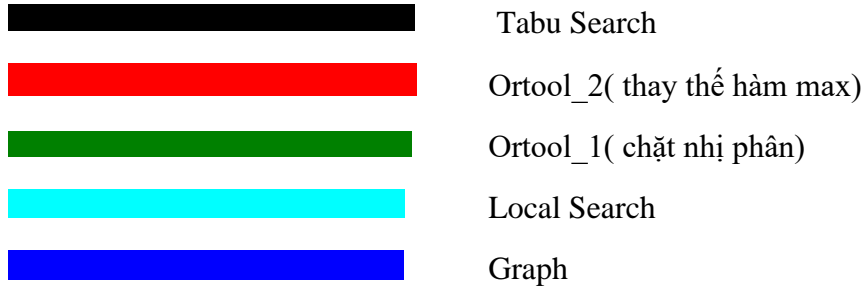
$$1 \leq N < 500, M = N + \lfloor N/5 \rfloor$$

$$100 \leq S[i].count \leq 200, 70 \leq C[i].count \leq 300$$

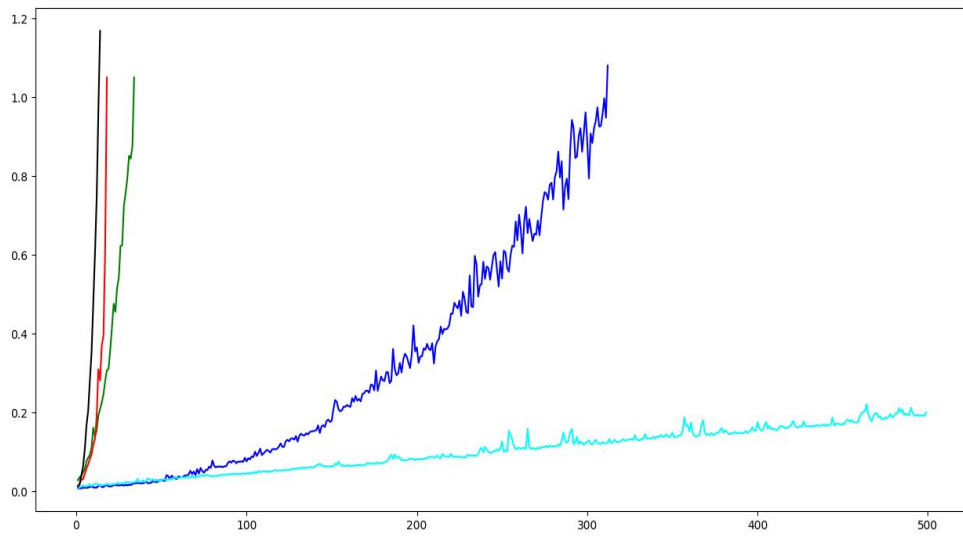
$$-1000 \leq S[i].x, S[i].y, C[i].x, C[i].y \leq 1000$$

VI, Đánh giá kết quả

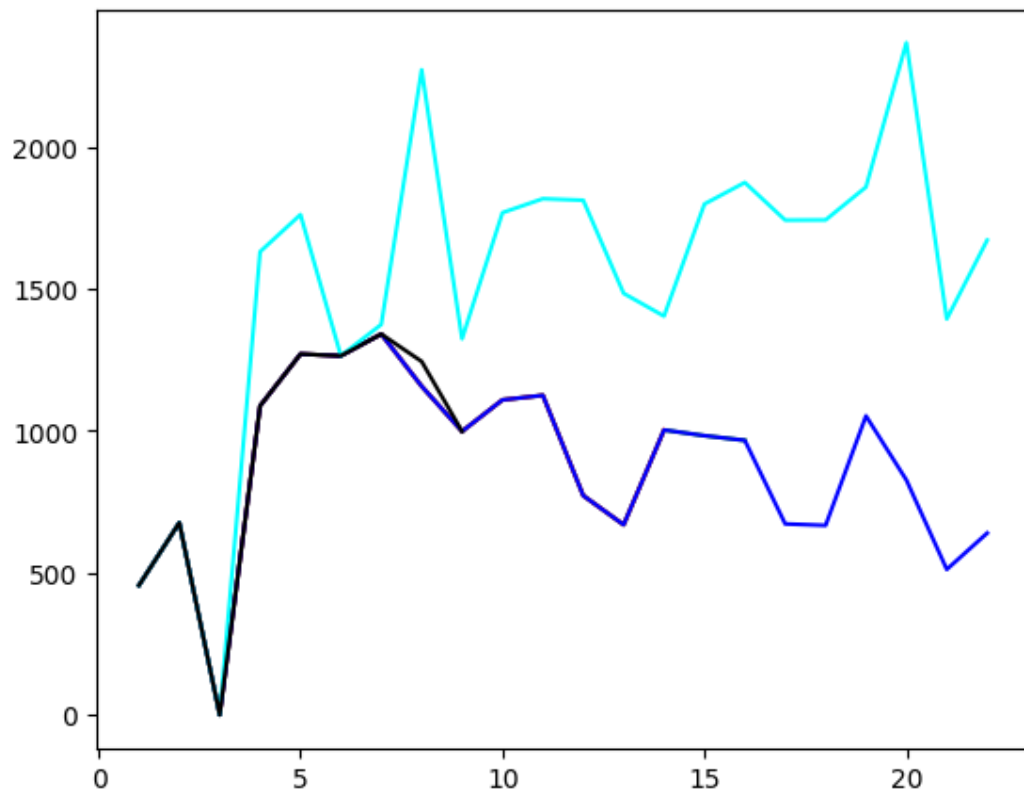
6.1 Trên bộ dữ liệu ngẫu nhiên:



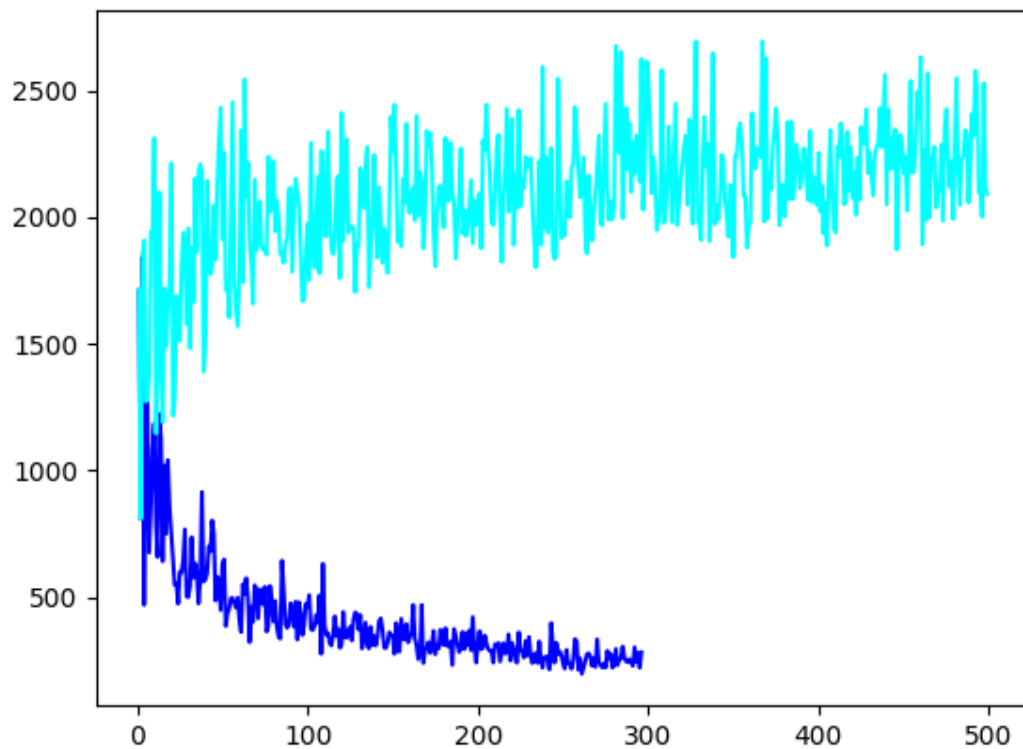
Đồ thị thời gian chạy của các thuật toán với $N \leq 23$



Đồ thị thời gian chạy của các thuật toán với $N < 500$



Đồ thị hàm mục tiêu của các thuật toán với $N < 23$



Đồ thị hàm mục tiêu chạy của thuật toán Local Search và Graph với $N < 500$

VII, Tài liệu tham khảo

<https://sites.google.com/site/kc97ble/algorithm-graph/cap-ghep-tren-do-thi-hai-phia-khong-trong-so>

<https://www.giaithuatlaptrinh.com/?p=2169>