

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
HỌC PHẦN: TRÍ TUỆ NHÂN TẠO – CT332
NGÀNH: KHOA HỌC MÁY TÍNH – KHÓA 46



Đề tài
THUẬT TOÁN DI TRUYỀN BÀI TOÁN
TÌM ĐƯỜNG ĐI NGƯỜI GIAO HÀNG

SINH VIÊN THỰC HIỆN:

NGUYỄN TUẤN ANH B2016945

NGUYỄN DUY BẢO B2016946

VÕ MINH KHANG B2016972

LÊ QUANG MINH B2016981

GVHD: VÕ TRÍ THỨC

MỤC LỤC

LỜI MỞ ĐẦU	3
PHẦN I: THUẬT TOÁN DI TRUYỀN	4
1. Giới thiệu	4
2. Nội dung	4
2.1. Cơ sở lý thuyết.....	4
2.1.1.Quá trình lai ghép (phép lai).....	4
2.1.2.Quá trình đột biến (phép đột biến):	4
2.1.3. Quá trình sinh sản và chọn lọc (phép tái sinh và phép chọn).....	5
2.2. Cấu trúc thuật toán di truyền tổng quát	5
PHẦN II : BÀI TOÁN TÌM ĐƯỜNG ĐI CỦA NGƯỜI GIAO HÀNG (Travelling Salesman Problem – TSP)	7
1. Giới thiệu	7
2. Phát biểu bài toán	7
PHẦN III: GIẢI THUẬT DI TRUYỀN BÀI TOÁN TÌM ĐƯỜNG ĐI NGƯỜI GIAO HÀNG	8
1. Giải thuật đề xuất:.....	8
2. Đồ thị đường đi người bán hàng.....	8
3. Khởi tạo quần thể:	8
4. Lai ghép	9
5. Đột biến	11
6. Di truyền:	12
7. Các hàm phụ	14
8. Đánh giá giải thuật và hướng phát triển	15
TÀI LIỆU THAM KHẢO	16

MỤC LỤC HÌNH ẢNH

Hình 1: Sơ đồ thuật toán di truyền.....	6
Hình 2: Đồ thị đường đi người bán hàng	8

LỜI MỞ ĐẦU

Trong ngành Khoa học máy tính, tìm kiếm lời giải tối ưu cho các bài toán là vấn đề đặc biệt rất quan tâm.

Mục đích chính của các thuật toán tìm kiếm lời giải là tìm ra lời giải tối ưu nhất cho bài toán trong thời gian nhỏ nhất. Các thuật toán như tìm kiếm không có thông tin hoặc vét cạn (tìm kiếm dựa trên danh sách, trên cây hoặc đồ thị) sử dụng phương pháp đơn giản và trực quan nhất hoặc các thuật toán tìm kiếm có thông tin heuristics để áp dụng các tri thức về cấu trúc của không gian tìm kiếm nhằm giảm thời gian cần thiết cho việc tìm kiếm được sử dụng nhiều nhưng chỉ với trong không gian tìm kiếm nhỏ và không hiệu quả trong không gian tìm kiếm lớn.

Tuy nhiên, trong thực tiễn có rất nhiều bài toán tối ưu với không gian tìm kiếm rất lớn cần phải giải quyết. Vì vậy, đòi hỏi các giải thuật có chất lượng cao và sử dụng kỹ thuật trí tuệ nhân tạo đặc biệt rất cần thiết khi giải quyết các bài toán có không gian tìm kiếm lớn. Thuật toán di truyền (Genetic Algorithm - GA) là một trong những kỹ thuật tìm kiếm lời giải tối ưu đã đáp ứng yêu cầu của nhiều bài toán và ứng dụng.

Hiện nay, thuật toán di truyền (GA) được ứng dụng rộng rãi trong các lĩnh vực phức tạp. Thuật toán di truyền (GA) chứng tỏ được hiệu quả của nó trong vấn đề khó có thể giải quyết bằng các phương pháp thông thường hay các phương pháp cổ điển, nhất là trong các bài toán cần có sự lượng giá, đánh giá sự tối ưu của kết quả thu được. Chính vì vậy, giải thuật di truyền đã trở thành đề tài nghiên cứu thú vị và đem đến nhiều ứng dụng trong thực tiễn.

Ngày nay, thuật toán di truyền (GA) được ứng dụng khá nhiều trong nhiều lĩnh vực khác nhau. Phải nói đến là các bài toán tối ưu bao gồm tối ưu số và tối ưu tổ hợp đã sử dụng GA để tìm lời giải cho bài toán người đi giao hàng (Traveling Salesman Problems – TSP). Thuật toán di truyền là một trong những thuật toán tối ưu vạch lộ trình khi di chuyển. Cho nên chúng em chọn đề tài: “Thuật toán di truyền ứng dụng của người đi giao hàng”.

PHẦN I: THUẬT TOÁN DI TRUYỀN

1. Giới thiệu

Thuật toán di truyền là thuật toán tối ưu ngẫu nhiên dựa trên mô phỏng quy luật đấu tranh sinh tồn của tự nhiên. Những sinh vật biến tiến hóa để thích nghi với hoàn cảnh sẽ tồn tại và phát triển. Thuật toán di truyền được ứng dụng đầu tiên trong hai lĩnh vực chính: tối ưu hóa và học tập của máy. Trong tối ưu hóa thuật toán di truyền được phát triển nhanh chóng và ứng dụng trong nhiều lĩnh vực khác nhau như tối ưu hàm, xử lý ảnh, bài toán hành trình người bán hàng, nhận dạng hệ thống và điều khiển. Thuật toán di truyền cũng như các thuật toán tiến hóa nói chung, hình thành dựa trên quan niệm cho rằng, quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất, hợp nhất và tự nó đã mang tính tối ưu. Quan niệm này có thể xem như một tiên đề đúng, không chứng minh được, không phù hợp với thực tế khách quan. Quá trình tiến hóa thể hiện tính tối ưu ở chỗ, không hẳn thế hệ sau bao giờ cũng tốt hơn thế hệ trước.

2. Nội dung

2.1. Cơ sở lý thuyết

Chọn mô hình (model, nhiễm sắc thể) để tượng trưng cho các giải pháp. Các mô hình có thể là dãy những số thập phân, nhị phân và có thể chạy hay hỗn hợp của chữ và số.

Thuật toán di truyền gồm có ba quy luật cơ bản là lai ghép, đột biến, sinh sản và chọn lọc tự nhiên

2.1.1. Quá trình lai ghép (phép lai)

Quá trình diễn ra bằng cách ghép một hay nhiều đoạn gen từ hai nhiễm sắc thể cha – mẹ để hình thành nhiễm sắc thể mới mang đặc tính của cha lẫn mẹ. Phép lai này có thể mô tả như sau: Chọn ngẫu nhiên hai hay nhiều cá thể trong quần thể. Giả sử chuỗi nhiễm sắc thể của cha và mẹ đều có chiều dài là m . Tìm điểm lai bằng cách tạo ngẫu nhiên một con số từ 1 đến $m - 1$. Như vậy, điểm lai này sẽ chia hai chuỗi nhiễm sắc thể cha – mẹ thành hai nhóm nhiễm sắc thể con là m_1 và m_2 . Hai chuỗi nhiễm sắc thể con lúc này sẽ là $m_{11} + m_{22}$ và $m_{21} + m_{12}$. Đưa hai chuỗi nhiễm sắc thể con vào quần thể để tiếp tục tham gia quá trình tiến hóa.

2.1.2. Quá trình đột biến (phép đột biến):

Đột biến là tình trạng nhiễm sắc thể con không có một (hoặc một số) tính trạng có trong mã di truyền của cha mẹ.

Các cặp cha mẹ được lựa chọn ngẫu nhiên và xác suất xảy ra đột biến với mỗi cặp được quy định từ trước, thường là rất nhỏ.

Các phép đột biến thường được sử dụng:

- Hoán vị: Đổi vị trí của các gen với nhau.
- Đổi giá trị: thay đổi giá trị tại một điểm gen.
- Đảo đoạn: Đảo thứ tự của một đoạn nhiễm sắc thể bất kì.

2.1.3. Quá trình sinh sản và chọn lọc (phép tái sinh và phép chọn)

Phép tái sinh: là quá trình các cá thể được sao chép dựa trên độ thích nghi của nó. Độ thích nghi là một hàm được gán các giá trị cho các cá thể trong quần thể của nó. Phép tái sinh có thể mô phỏng như sau: Tính độ thích nghi của từng cá thể trong quần thể, lập bảng cộng dồn các giá trị thích nghi đó (theo thứ tự gán cho từng cá thể) ta được tổng độ thích nghi. Giả sử quần thể có n cá thể. Gọi độ thích nghi của cá thể i là F_i , tổng dồn thứ i là F_t . Tổng độ thích nghi là F_m . Tạo ngẫu nhiên F có giá trị trong đoạn từ 0 đến F_m . Chọn cá thể k đầu tiên thỏa mãn $F \geq F_t$ đưa vào quần thể của thế hệ mới.

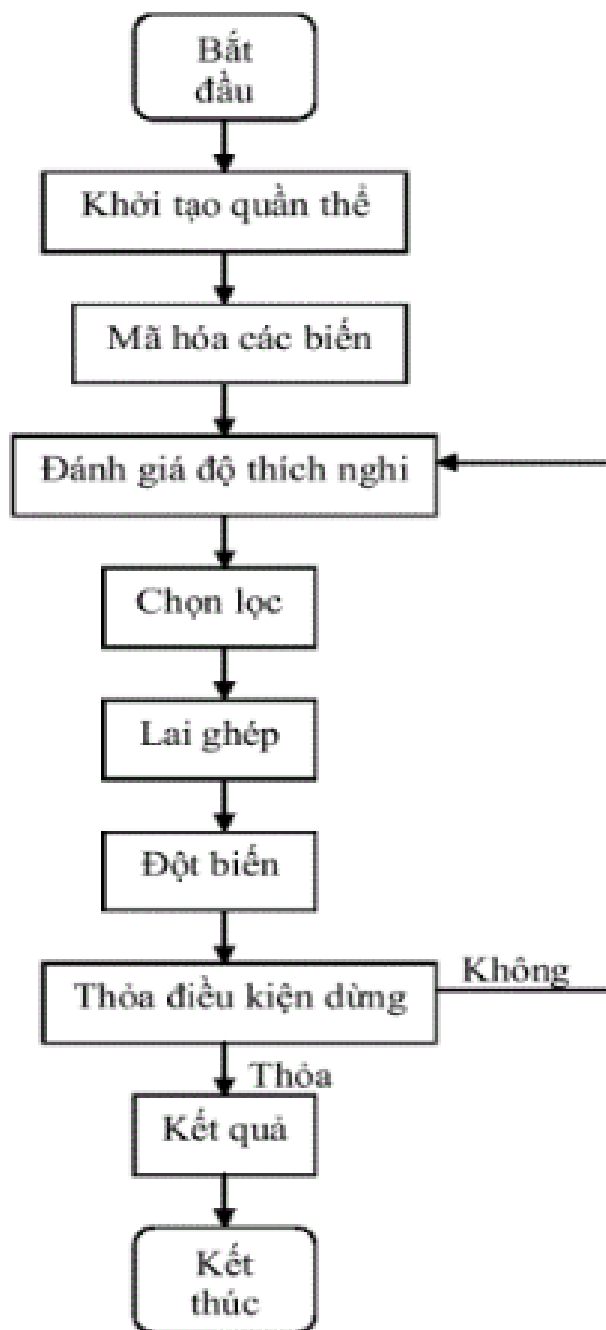
Phép chọn: là quá trình loại bỏ các cá thể xấu và để lại những cá thể tốt. Phép chọn được mô tả như sau: sắp xếp quần thể theo thứ tự độ thích nghi giảm dần loại bỏ các cá thể cuối dãy, chỉ để lại n cá thể tốt nhất.

2.2. Cấu trúc thuật toán di truyền tổng quát

Thuật toán di truyền bao gồm các bước sau:

- Bước 1: Khởi tạo quần thể các nhiễm sắc thể.
- Bước 2: Xác định giá trị thích nghi của từng nhiễm sắc thể.
- Bước 3: Sao chép lại nhiễm sắc thể dựa vào giá trị thích nghi của chúng ta tạo ra những nhiễm sắc thể mới bằng các phép toán di truyền.
- Bước 4: Loại bỏ những thành viên không thích nghi trong quần thể.
- Bước 5: Chèn những nhiễm sắc thể mới vào quần thể để hình thành một quần thể mới.
- Bước 6: Nếu mục tiêu tìm kiếm đạt được thì dừng lại, nếu không trở lại bước 3.

Sơ đồ thuật toán :



Hình 1: Sơ đồ thuật toán di truyền

PHẦN II : BÀI TOÁN TÌM ĐƯỜNG ĐI CỦA NGƯỜI GIAO HÀNG (Travelling Salesman Problem – TSP)

1. Giới thiệu

Bài toán người bán hàng (travelling salesman problems – TSP) là một bài toán thuộc thể loại tối ưu tổ hợp được nghiên cứu trong lý thuyết khoa học máy tính. Nội dung bài toán có thể hiểu khái quát như sau: cho trước một danh sách các thành phố và khoảng cách giữa chúng, tìm đường đi ngắn nhất đi qua tất cả các thành phố đúng 1 lần.

Bài toán là một trong những bài toán được nghiên cứu sâu nhất trong tối ưu hóa. Nó thường dùng làm thước đo cho nhiều phương pháp tối ưu hóa. Mặc dù bài toán rất khó giải trong trường hợp tổng quát, có nhiều phương pháp giải chính xác cũng như heuristic đã được tìm ra để giải quyết một số trường hợp có tới hàng chục nghìn thành phố.

Ngay trong hình thức phát biểu đơn giản nhất, bài toán TSP đã có nhiều ứng dụng trong lập kế hoạch, hậu cần, cũng như thiết kế vi mạch,...Đến nay bài toán TSP vẫn được nghiên cứu tìm ra lời giải cho các bộ dữ liệu lớn hơn.

2. Phát biểu bài toán

Phát biểu bài toán: Cho đồ thị đầy đủ n đỉnh vô hướng, có trọng số $G = (V, E)$. Tìm đường đi $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n \rightarrow v_1$ với $v_i \in V, i = \overline{1, n}$ sao cho tổng trọng số hành trình trên các cạnh (v_i, v_{i+1}) và (v_n, v_1) là nhỏ nhất.

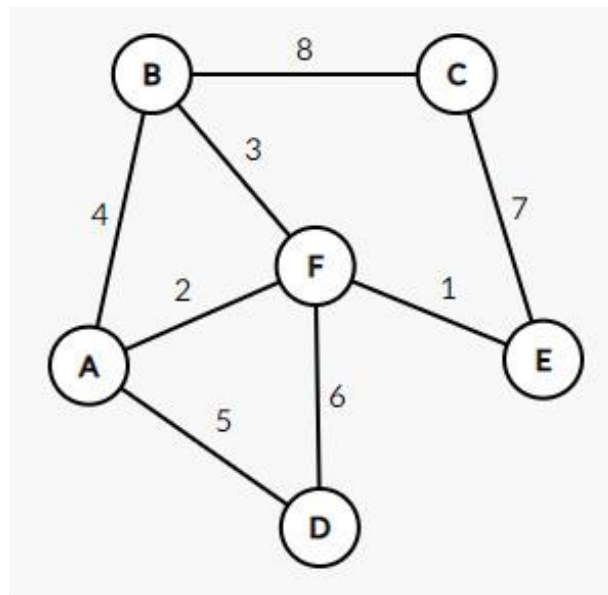
PHẦN III: GIẢI THUẬT DI TRUYỀN

BÀI TOÁN TÌM ĐƯỜNG ĐI NGƯỜI GIAO HÀNG

1. Giải thuật đề xuất:

Nhóm đề xuất giải thuật di truyền để giải bài toán người hàng. Giải thuật được cài đặt bằng ngôn ngữ C.

2. Đồ thị đường đi người bán hàng



Hình 2: Đồ thị đường đi người bán hàng

3. Khởi tạo quần thể:

Quần thể ban đầu được tạo ra bằng cách sinh ngẫu nhiên các đường đi khởi tạo là một nửa số kích thước cá thể tối đa. Việc sinh ngẫu nhiên sử dụng hàm đột biến. Số kích thước cá thể tối đa có thể tùy biến theo số đỉnh của đồ thị cần giải, sau nhiều lần chạy nhóm chọn kích thước quần thể là 100 cá thể.

4. Lai ghép

Phương thức lai ghép được thực hiện dựa trên 2 cá thể đầu vào C1 và C2:

Ví dụ như sau:

C1	E	A	C	B	D	F
----	---	---	---	---	---	---

C2	A	C	B	E	F	D
----	---	---	---	---	---	---

Thực hiện lai ghép 1 điểm cắt với vị trí ngẫu nhiên:

Truyền lần lượt đường của C1 và C2 qua Cr1 và Cr2 để giữ nguyên đường đi C1 và C2.

Lấy một ví dụ điểm cắt $p = 4$.

- Cắt từ điểm đầu của Cr1 đến p và đưa vào một cá thể tạm (temp)

Cr1					D	F
-----	--	--	--	--	---	---

temp	E	A	C	B		
------	---	---	---	---	--	--

- Cắt từ điểm đầu của Cr2 và đưa vào đoạn bị cắt của Cr1

Cr1	A	C	B	E	D	F
-----	---	---	---	---	---	---

Cr2					F	D
-----	--	--	--	--	---	---

- Đưa đoạn đường đi đang có trong cá thể tạm (temp) vào C2:

Cr1	A	C	B	E	D	F
-----	---	---	---	---	---	---

Cr2	E	A	C	B	F	D
-----	---	---	---	---	---	---

Ta thu được 2 con Cr1 và Cr2 là con của C1 và C2 sau khi đã lai ghép và tính lại hệ số thích nghi cho 2 đường đi mới sinh.

Ta trả về 1 danh sách chứa 2 con Cr1 và Cr2 để tiếp tục quá trình đột biến.

Lưu ý: Một số trường hợp sau khi lai ra 2 con Cr1 và Cr2 sẽ mắc vào trường hợp thành phố bị lặp. Ví dụ như E A C B F C, trong trường hợp này hệ số thích nghi được tính rất lớn, ta cũng sẽ tiếp tục quá trình đột biến với hy vọng sẽ thay đổi thành số đường đi không lặp.

Cài đặt code:

```
List lai_ghep (CaThe C1, CaThe C2){
    int i;
    CaThe cr1 = c1;
    CaThe cr2 = c2;
    List l;
    init_List(&l);
    int index = ngau_nhien(0,sotp-1);
    int temp[index+1];
    //Lai giua ca the cr1 va ca the cr2
    for (i=0; i<=index; i++)
        temp[i] = cr1.A[i];
    for (i=0; i<=index; i++)
        cr1.A[i] = cr2.A[i];
    for (i=0; i<=index; i++)
```

```

        cr2.A[i] = temp[i];

        cr1.he_so_thich_nghi = tinh_he_so_thich_nghi(cr1.A);
        cr2.he_so_thich_nghi = tinh_he_so_thich_nghi(cr2.A);

        //Them 2 ca the moi vao ds
        append_List(&l,cr1);
        append_List(&l,cr2);

        return L;
    }

```

5. Đột biến

Phương thức đột biến được dựa trên 1 cá thể đầu vào:

C1	A	E	C	B	D	F
----	---	---	---	---	---	---

Thực hiện đột biến bằng cách biến đổi một điểm gen ngẫu nhiên bất kỳ dựa trên một đường đi. Số lần đột biến sẽ được chọn ngẫu nhiên trong một khoảng giá trị nhất định.

Cách làm này sẽ có 2 khả năng:

- Biến đường đi hợp lệ thành không hợp lệ (1)
- Biến đường đi không hợp lệ thành hợp lệ (2)

Vì bản chất của phép đột biến là có thể sinh ra cá thể mang tính trạng thái xấu nên khả năng (1) là không tránh khỏi.

Cài đặt code:

```
void dot_bien(CaThe *c){
    int index = ngau_nhien(0, sotp-1);
    int count = ngau_nhien(1, 10);
    int value;
    while(count > 0){
        value = ngau_nhien(1, sotp);
        c->A[index] = value;
        count--;
    }
    c->he_so_thich_nghi = tinh_he_so_thich_nghi(c->A);
}
```

6. Di truyền:

Kích thước quần thể là cố định qua các thế hệ. Ở mỗi thế hệ ta lại có các cá thể mới sinh bằng lai ghép và đột biến do đó cần phải có sự chọn lọc để đảm bảo tính cân bằng của quần thể cũng chính là tránh các lỗi phát sinh về bộ nhớ khi kích thước quần thể quá lớn.

Sau khi lai qua mỗi thế hệ, ta sẽ sử dụng hàm Sort để sắp xếp độ thích nghi của các cá thể theo thứ tự tăng dần, và mặc định sẽ gán cho cá thể đầu tiên là cá thể tốt nhất. Nếu cá thể đầu tiên của quần thể sau n thế hệ lại không tốt hơn cá thể đầu tiên của thế hệ n-1 thì ta vẫn sẽ giữ cá thể của lần lai thứ n-1. Với mỗi lần lai ta sẽ cần 1 cặp cha và mẹ, do vậy quần thể sẽ có n/2 cặp lai và sau khi lai thì vẫn đảm bảo số cá thể không đổi.

Hàm di truyền này sẽ trả về 1 cá thể có độ thích nghi tốt nhất (chi phí nhỏ nhất) sau khi lai qua n thế hệ và tốt nhất trong số cá thể của quần thể chứa nó.

Cài đặt code:

```
CaThe di_truyen(int slct, int TheHe){
    int i, j;
    QuanThe qt;
    CaThe tot;
    khoi_tao_quan_the(&qt,slct);
    sap_xep_quan_the(&qt);
    tot = qt.CT[0]; //Mac dinh ca the dau tien la tot nhat
    for (i=1; i<=TheHe; i++){
        sap_xep_quan_the(&qt);
        if (qt.CT[0].he_so_thich_nghi < tot.he_so_thich_nghi)
            tot = qt.CT[0];
        QuanThe new;
        init_QT(&new,slct);
        for (j=0; j<slct; j+=2){
            int cha, me;
            while (1){
                cha = ngau_nhien(0, slct-1);
                me = ngau_nhien(0, slct-1);
                if (cha != me)
                    break;
            }
            List l = lai(qt.CT[cha], qt.CT[me]);
            dot_bien(&l.E[0]);
            dot_bien(&l.E[1]);
            new.CT[j] = l.E[0];
            new.CT[j+1] = l.E[1];
        }
    }
}
```

```

    }
    qt = new;    //Quan the New cung se co 100 con
}
return tot;
}

```

Lưu ý: Do được xây dựng trong khoảng thời gian ngắn nên hàm này vẫn chưa được tối ưu, sẽ có một số cá thể lai nhiều lần và cũng sẽ có một số cá thể không lai được lần nào trong sau khi qua 1 thế hệ, nhưng xác suất này xảy ra rất nhỏ khoảng 2% trong 1 thế hệ, nên ta có thể an tâm sau khi di truyền qua một số lần nhất định thì tất cả các cá thể trong quần thể sẽ được lai với nhau.

7. Các hàm phụ

int ngau_nhien (int from, int to): Trả về 1 số ngẫu nhiên trong phạm vi từ from đến to.

int lap_phan_tu (int a[]): Trả về tổng các lần lặp của tất cả phần tử trong mảng.

int tinh_he_so_thich_nghi (int a[]): Trả về độ dài đường đi của các thành phố trong mảng a[].

void khoi_tao_ca_the (CaThe *c): Khởi tạo thành phố .

void hien_thi_ca_the(CaThe c): Hiện thị thành phố.

void khoi_tao_quan_the (QuanThe *qt, int n): Khởi tạo quần thể với n cá thể.

void sap_xep_quan_the (QuanThe *qt): Sắp xếp quần thể dựa trên hệ số thích nghi.

8. Đánh giá giải thuật và hướng phát triển

Giải thuật đã đề xuất đáp ứng cơ bản các bước trong giải thuật di truyền. Kết quả chạy giải thuật cho ra kết quả tối ưu trong các trường hợp số đỉnh nhỏ. Còn với những trường hợp số đỉnh lớn vẫn chưa tìm được lời giải.

Giải thuật di truyền mà nhóm làm phần lớn dữ liệu đưa vào là dữ liệu vào phải có đường đi và chưa tìm được đường đi có chu trình như kỳ vọng. Nhóm sẽ tìm hiểu và tối ưu giải thuật để khắc phục các khuyết điểm trên và cố gắng tìm ra chu trình đường đi ngắn nhất với số thể hệ cũng như số cá thể ở mức tối thiểu nhất.

TÀI LIỆU THAM KHẢO

https://vi.wikipedia.org/wiki/B%C3%A0i_to%C3%A1n_ng%C6%B0%E1%BB%9Di_b%C3%A1n_h%C3%A0ng

<https://hotroontap.com/bai-tap-lon-tri-tue-nhan-cao-tim-hieu-giai-thuat-di-truyen/>

https://www.academia.edu/5015312/Tu%E1%BA%A7n_10_H%E1%BB%8Dc_li%E1%BB%87u_%E1%BB%A8ng_d%E1%BB%A5ng_thu%E1%BA%ADt_gi%E1%BA%A3i_GA_%C4%91%E1%BB%91i_v%E1%BB%9Bi_b%C3%A0i_to%C3%A1n_du_l%E1%BB%8Bch_tham_kh%E1%BA%A3o

<https://viblo.asia/p/thuat-toan-di-truyen-ung-dung-giai-mot-so-bai-toan-kinh-dien-phan-2-07LKXwYp5V4>