

LUYỆN TẬP THIẾT KẾ THUẬT TOÁN - NHÓM 4

LƯƠNG TRIỀU VỸ
DANH VÕ HỒNG PHÚC

GIỚI THIỆU

ÔN TẬP KIẾN THỨC

01

03

MỞ RỘNG VÀ ỨNG DỤNG

LUYỆN TẬP

02

04

TỔNG KẾT NHẬN XÉT



1. ÔN TẬP

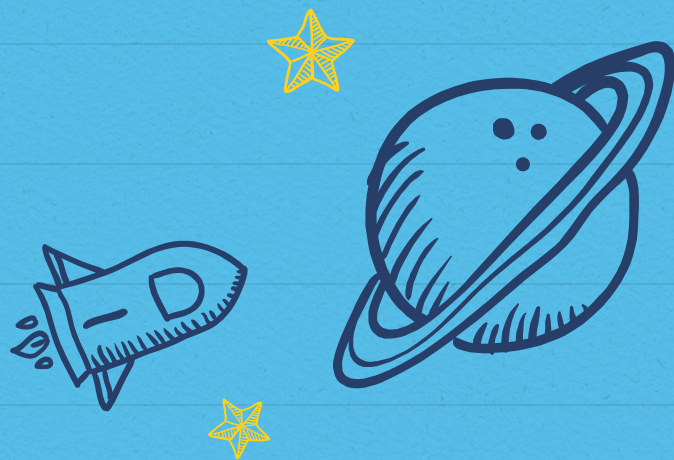
LƯỢT CHƠI

CÁC ĐỘI TĂNG ĐIỂM BẰNG CÁCH THAM GIA PHÁT BIỂU, TRẢ LỜI CÂU HỎI

CÁC CÂU TRẢ LỜI ĐÚNG SẼ ĐƯỢC CỘNG 10 ĐIỂM, CÒN LẠI SẼ ĐƯỢC CỘNG 5 ĐIỂM

1 PHẦN THƯỞNG SẼ TRAO CHO 3 ĐỘI CÓ SỐ ĐIỂM CAO NHẤT VÀO CUỐI GIỜ

"AN ALGORITHM MUST BE SEEN TO BE
BELIEVED"



2. LUYỆN TẬP



NHÂN VẬT

X Huy là sinh viên năm cuối ngành cơ khí tại BK, vì thích CNTT nên đã chuyển ngang sang ngành KHMT tại UIT, trong quá trình học, Huy tìm được cuốn 300 bài code thiếu nhi, tuy nhiên bài tập trong cuốn này quá khó, Huy mong các bạn có thể giúp Huy hoàn thành hết số bài tập trong cuốn sách này.





- X Cho 3 điểm A,B,C trong không gian ba chiều. Tính khoảng cách từ đường thẳng A đến đường thẳng BC.
- X Đề bài bao gồm T trường hợp ($1 \leq T \leq 10^4$), mỗi trường hợp gồm 1 hàng gồm 9 số nguyên cách nhau lần lượt là tọa độ của 3 điểm A,B,C

BẠN CÓ BIẾT ?



Gennady Korotkevich (Tourist) người ngoài hành tinh trong giới Thuật Toán:

- Biết viết phần mềm đầu tiên vào năm 8 tuổi
- Tham gia kì thi IOI vào năm 11 tuổi, sau đó 6 lần dành HCV IOI trong liên tiếp 6 năm
- 2 lần vô địch kì thi ACM-ICPC World Finals năm 2013 và 2015
- Anh cũng nhiều lần vô địch các kì thi Thuật toán do Facebook, Google tổ chức,...



LỜI GIẢI



Abstraction:

- Tìm khoảng cách từ 1 điểm đến 1 đường thẳng cho trước

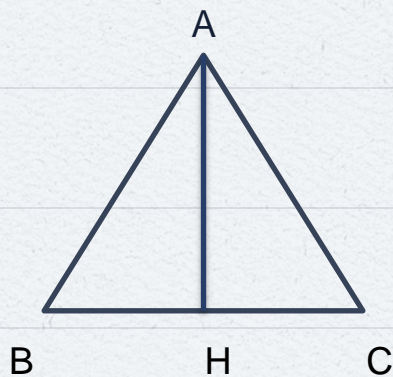
Decompositon:

- Thay vì giải bài toán trong hệ tọa độ Oxyz, ta đưa bài toán về hệ tọa độ Oxy

Pattern Regconition:

Đây là một bài thuần toán, ta vẽ tam giác ABC từ 3 điểm, gọi :

- $a = BC$, $b = AC$, $c = AB$
- P : nửa chu vi tam giác ABC, S : diện tích tam giác,
- $S = \sqrt{p(p-a)(p-b)(p-c)}$
- Gọi $h = AH$ là đường cao kẻ từ đỉnh A tới cạnh BC ta có $S = \frac{1}{2} \cdot BC \cdot AH$
 $= \frac{1}{2} \cdot a \cdot h$
- Suy ra $h = \frac{2 \cdot S}{a}$



Algorithm Design:

- Áp dụng công thức tính khoảng cách cho 2 điểm trong không gian ba chiều.
- Áp dụng công thức tính diện tích, chu vi tam giác. Từ đó ta tìm được h.

Evaluation:

- Độ phức tạp: $O(1)$

```
int T;
double xa,ya,za,xb,yb,zb,xc,yc,zc;

double Dist(double xa,double ya,double za,double xb,double yb,double zb)
{
    return sqrt((xa-xb)*(xa-xb)+(ya-yb)*(ya-yb)+(za-zb)*(za-zb));
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin>>T;
    while(T--)
    {
        cin>>xa>>ya>>za;
        cin>>xb>>yb>>zb;
        cin>>xc>>yc>>zc;
        double c=Dist(xa,ya,za,xb,yb,zb);
        double b=Dist(xa,ya,za,xc,yc,zc);
        double a=Dist(xc,yc,zc,xb,yb,zb);
        double p=(a+b+c)/2.0;
        double S=sqrt(p*(p-a))*sqrt((p-b)*(p-c));
        double h=2.0*S/a;
        cout<<setprecision(2)<<fixed<<h<<"\n";
    }
}
```

Code tham khảo bằng python:

```
import math

def distance(xa,ya,za,xb,yb,zb):
    return math.sqrt((xa-xb)*(xa-xb)+(ya-yb)*(ya-yb)+(za-zb)*(za-zb))

T = int(input())

for i in range(0,T):
    xa,ya,za,xb,yb,zb,xc,yc,zc = map(float,input().split())
    c = distance(xa,ya,za,xb,yb,zb)
    b = distance(xa,ya,za,xc,yc,zc)
    a = distance(xc,yc,zc,xb,yb,zb)
    p = (a+b+c)/2.0
    S = math.sqrt(p*(p-a))*math.sqrt((p-b)*(p-c))
    h = 2.0*S/a
    print(h)
```


BÀI 2

Sau khi hoàn thành 12 bài đầu của cuốn sách, Huy đã vượt qua được bài thi cuối kì Nhập môn lập trình với điểm số khá cao. Để cảm ơn bạn, Huy đố bạn một bài toán khá hay mà Huy đang làm dở trong cuốn sách. Đề bài như sau:

Cho 1 cây có N đỉnh ($1 \leq N \leq 10^5$), đỉnh được đánh số từ 1 tới N . Cây có $N-1$ cạnh, trong đó cạnh thứ i nối giữa đỉnh x_i và y_i . Bạn sẽ tô màu mỗi đỉnh trên cây bằng 1 trong 2 màu đen trắng sao cho không tồn tại 2 đỉnh kề mà 2 đỉnh đều có màu đen.

Tìm số cách tô màu các đỉnh thỏa mãn, modulo 10^9+7

Input:

- N
- $N-1$ dòng gồm 2 số nguyên x_i và y_i

Output:

- Số cách tô màu thỏa mãn



BẠN CÓ BIẾT ?

Mạnh Luật - Cựu sinh viên UIT

- CEO và đồng sáng lập của Cyber Jutsu.
- Cựu Kỹ sư phần mềm của Microsoft, MSRC, trụ sở Anh Quốc,
- Cựu Chuyên viên Nghiên cứu Bảo mật của Tencent
- Cựu Kỹ sư bảo mật ứng dụng của tập đoàn VNG
- Quán quân cuộc thi Cyber SEA Game 2015.
- Top 3 cuộc thi DEFCON CTF 2017 - đây được xem như là giải vô địch CTF toàn thế giới khi chỉ có 16 đội lọt vào vòng chung kết.



MR. MANH LUAT

CEO & Co-founder Cyber Jutsu



LỜI GIẢI



Abstraction:

- Cho cây gồm N đỉnh, có $N-1$ cạnh, đếm số cách tô màu các đỉnh sao cho thỏa mãn yêu cầu đề bài

Decomposition:

- Chọn một đỉnh là gốc, ta chia nhỏ bài toán từ tìm tất cả các cách tô màu đỉnh gốc thành tìm tất cả các cách tô màu các con của gốc

Partern Regconition:

- Từ gốc để tìm tất cả các cách tô màu từ các nút con
- => Quy hoạch động trên cây.



Algorithm Design:

- Đầu tiên, chọn một đỉnh bất kỳ làm gốc, gọi đỉnh đó là root. Gọi $dp[i][j]$ là số cách để tô màu cây con của đỉnh i , với đỉnh i được tô màu j .
- Chúng ta sẽ coi $j=0$ là màu đen và $j=1$ là màu trắng.
- Nếu chúng ta tô đỉnh i màu đen, thì toàn bộ các con của nó phải màu trắng. Do đó, ta có:
$$dp[i][0] = \prod dp[c][1]$$
 với mọi c thuộc tập con của i
- Nếu chúng ta tô đỉnh i màu trắng, thì các con của nó có thể có màu đen hoặc trắng đều được. Do đó, ta có:
$$dp[i][1] = \prod (dp[c][1] + dp[c][0])$$
- Đáp án sẽ là $dp[root][0] + dp[root][1]$

Evaluation:

- Độ phức tạp: $O(n)$

```
const int MAXN = 1e5;
const int MOD = 1e9 + 7;

int add(int i, int j) {
    if ((i += j) >= MOD)
        i -= MOD;
    return i;
}

int mul(int i, int j) {
    return int((long long) i * j % MOD);
}

vector<int> adj[MAXN];
int dp[MAXN][2];

void dfs(int i, int p) {
    dp[i][0] = dp[i][1] = 1;
    for (int j : adj[i]) if (j != p) {
        dfs(j, i);
        dp[i][0] = mul(dp[i][0], dp[j][1]);
        dp[i][1] = mul(dp[i][1], add(dp[j][0], dp[j][1]));
    }
}

int main() {
    cin.tie(0) -> sync_with_stdio(0);

    int n;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        int a, b;
        cin >> a >> b;
        --a;
        --b;
        adj[a].push_back(b);
        adj[b].push_back(a);
    }
    dfs(0, 0);
    cout << add(dp[0][0], dp[0][1]) << '\n';
}
```


Code tham khảo

```
MOD = 10 ** 9 + 7
N = int(input())
graph = [[] for _ in range(N + 1)]

for i in range(N - 1):
    x, y = map(int, input().split())
    graph[x].append(y)
    graph[y].append(x)

dp = [[1] * 2 for _ in range(N + 1)]

visited = [False] * (N + 1)

def dfs(now, prev):
    if visited[now]:
        return
    visited[now] = True

    for next in graph[now]:
        if visited[next]:
            continue
        dfs(next, now)
        dp[now][0] *= dp[next][0] + dp[next][1]
        dp[now][0] %= MOD
        dp[now][1] *= dp[next][0]
        dp[now][1] %= MOD

dfs(1, -1)

ans = (dp[1][0] + dp[1][1]) % MOD
print(ans)
```

BÀI 3

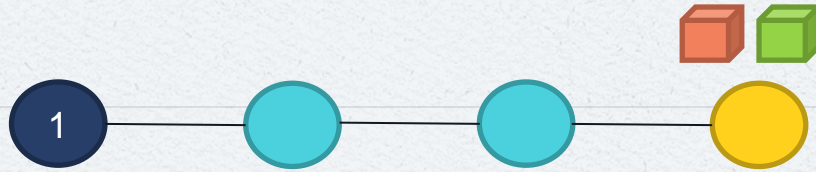
Sau khi giải được các bài code thiếu nhi, Huy thấy trình mình lên hẳn và muốn thử sức với các bài toán thực tế. Một hôm Huy lang thang trên facebook và thấy bài toán mà Chính phủ đang đau đầu giải quyết. Bài toán như sau:

Vùng đồng bằng sông Hồng có N thành phố, được nối với nhau bằng m đoạn đường 2 chiều. Do tình hình COVID-19 nên chính phủ muốn vận chuyển lương thực từ các thành phố ra bến cảng để trợ giúp đồng bào tại TP.HCM. Bến cảng nằm ở thành phố 1. Các thành phố còn lại, mỗi thành phố đều có một kho lương thực. Chính phủ muốn đẩy nhanh quá trình vận chuyển nhanh nhất nhưng vẫn đảm bảo an toàn nên thực hiện theo cách sau:

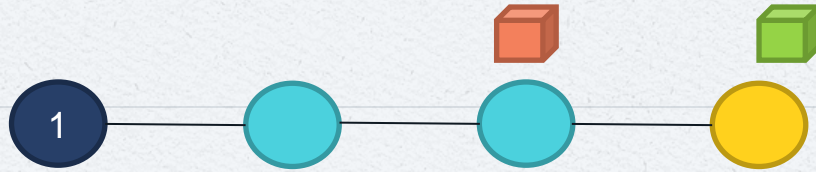
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



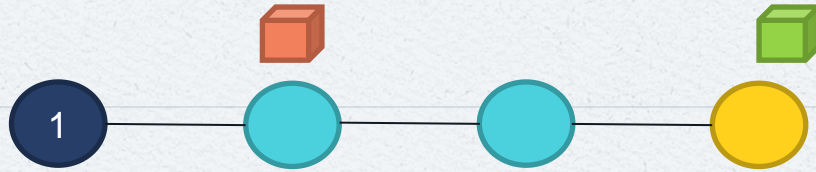
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



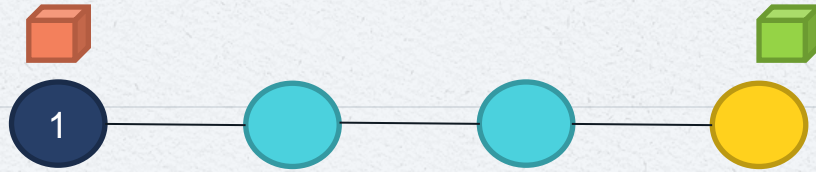
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



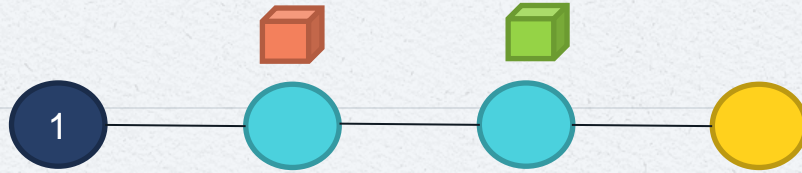
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



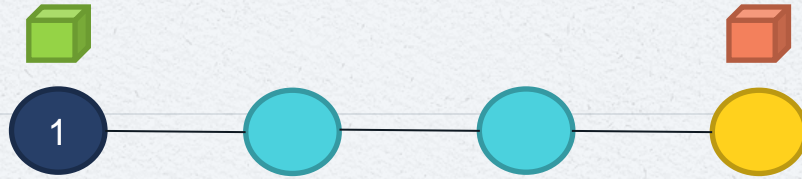
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



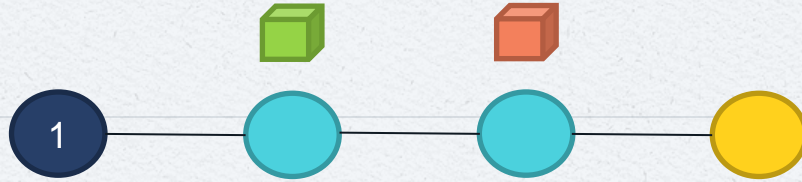
Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian



Ban đầu, tại nhà kho ở mỗi thành phố, có 2 xe dùng để vận chuyển lương thực. Sau khi một xe đã vận chuyển tới cảng an toàn, xe kia được bắt đầu xuất phát, xe chở hàng xong lập tức quay về kho để tiếp tục chuyển hàng. Thời gian di chuyển giữa 2 thành phố là 1 đơn vị thời gian

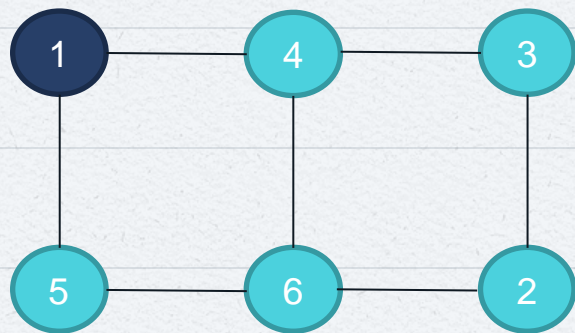


BÀI 3

Yêu cầu: Để viện trợ đủ lương thực cho TP.HCM cần k xe xuất ra bến cảng. Hãy xác định thời gian sớm nhất để Chính phủ hoàn thành kế hoạch

Dữ liệu:

- Dòng đầu chứa 3 số nguyên dương n,m,k ($m, n \leq 10^5$; $k \leq 10^{15}$)
 - m dòng tiếp theo, mỗi dòng gồm 2 số u,v xác định đoạn đường nối 2 thành phố u và v.
- Dữ liệu đảm bảo các thành phố đều có đường đi đến thành phố 1



INPUT	OUTPUT
6 7 10 1 4 5 1 6 5 4 6 4 3 2 3 2 6	4

Ở thời điểm 1, có 02 xe từ thành phố 4,5 tới. Thời điểm 2, có 04 xe từ các thành phố 3, 4, 5, 6 tới cảng. Thời điểm 3, có 03 xe từ các thành phố 1, 4, 5 tới. Thời điểm 4, chỉ cần thêm 01 xe từ một trong các thành phố 3, 4, 5, 6 tới sẽ đủ 10 xe

LỜI GIẢI

Decomposition

- Từ một thành phố, luôn đi đường ngắn nhất đến 1
- Tại một đỉnh cách đỉnh 1 khoảng cách d . Đến thời điểm T sẽ có T/d xe đã vận chuyển được hàng đến 1

Abstraction

- Input: Cho đồ thị N đỉnh, M cạnh và số K
- Output: Tìm T nhỏ nhất sao cho tổng các $T/d[i] \geq K$

Pattern Recognition

- Tìm đường đi ngắn nhất từ một thành phố nhỏ đến đỉnh 1 BFS
- Tìm kiểm nhị phân để xác định T



Algorithm Design

- Dùng BFS tạo mảng d , với $d[i]$ là khoảng cách nhỏ nhất từ đỉnh i đến đỉnh 1
- Tìm kiểm nhị phân T trong khoảng $0 \rightarrow 1e18$:

for i in $(2, n)$:

$sum = sum + T/d[i]$

if $(sum \geq K) \rightarrow$ giảm T



EVALUATE

BFS:

$$O(N+M)$$

Binary Search:

$$O(\log(K))$$

$$\rightarrow \max(O(N+M), O(\log(K)))$$

$$N, M \leq 10^5$$

$$K \leq 10^{18}$$



CODE MẪU

```
12 def BFS():
13     d[1] = 0
14     queue = []
15     queue.append(1)
16     while queue:
17         u = queue.pop(0)
18         for v in edge[u]:
19             if d[v] == -1:
20                 d[v] = d[u] + 1
21                 q.append(v)
22
23 def check(T):
24     sum = 0
25     for i in range(2,n):
26         if (d[i] > 0):
27             sum += T/d[i]
28             if sum >= k:
29                 return 1
30     return 0
31
32 l = 0
33 r = 10**18
34 while (r-l>1):
35     mid = int((l+r) / 2)
36     if check(mid):
37         r = mid
38     else:
39         l = mid
```

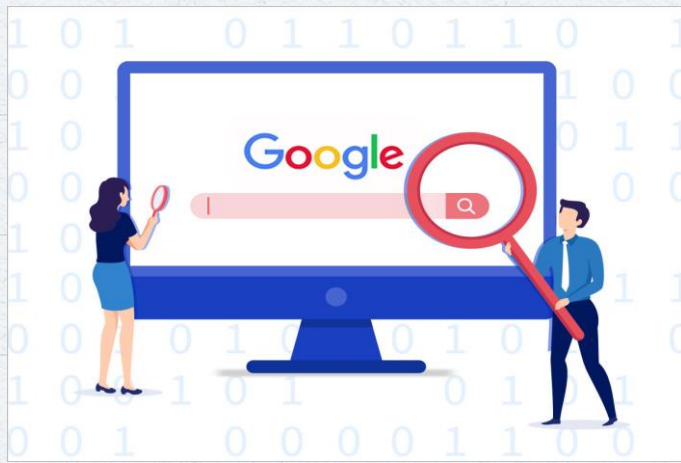

Bẵng đi một thời gian, Huy hiện giờ đã là một kỹ sư phần mềm chuyên nghiệp lâu năm trong một tập đoàn lớn với mức lương hơn 300 triệu/tháng. Huy nhớ lại cuốn sách ngày xưa vẫn còn chưa giải hết nên cố gắng tìm lại, tuy nhiên thì cuốn sách đó đã thất lạc. Tương truyền bất cứ ai giải được hết tất cả các bài tập trong cuốn sách đó thì trên thông thiên văn, dưới tường địa lí, coi như có sẵn một vị trí trong FAANG. Cuốn sách 300 bài code thiếu nhi ngày qua ngày vẫn đang chờ đợi người xứng đáng tìm được. Chúc bạn là người may mắn đó

THE END



ỨNG DỤNG





- Ví dụ, có 2 chuỗi "AABAA" và "AAAB". Để chuyển chuỗi đầu thành chuỗi sau, đơn giản nhất là xóa ký tự 'B' ở giữa rồi chuyển 'A' cuối chuỗi thành 'B'. Thuật toán này có rất nhiều ứng dụng, ví dụ như trong các vấn đề liên quan đến DNA hay chống đạo văn. Với các lập trình viên, thuật toán này thường được dùng trong việc so sánh 2 phiên bản source code (mã nguồn) của cùng 1 file. Nếu các phần tử của chuỗi là các dòng của file, thuật sẽ cho ta biết dòng code nào bị xóa, dòng nào bị thêm vào hay sửa đi trong các phiên bản đó.
- Không có quy hoạch động, ta sẽ phải xét các trường hợp theo cấp số mũ để biến một chuỗi thành chuỗi khác. Với quy hoạch động, bài toán này được giải quyết với độ phức tạp chỉ $O(N \times M)$, trong đó N và M là số phần tử của mỗi chuỗi.

ROADMAP

Start learning a
language first

Basics Algorithms

Build real life projects

1

3

5

2

4

6

Complexities

Advance
Topics

Keep Practicing



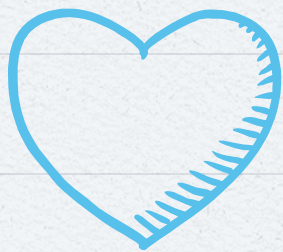
EXTRA RESOURCES

[HTTPS://WWW.HACKEREARTH.COM/](https://www.hackerearth.com/)

[HTTPS://CP-ALGORITHMS.COM/](https://cp-algorithms.com/)

[HTTPS://CODEFORCES.COM/BLOG/ENTRY/23054](https://codeforces.com/blog/entry/23054)

[HTTPS://OJ.VNOI.INFO/](https://oj.vnoi.info/)



THANKS!

**Có câu hỏi hoặc thắc mắc ?
Hãy liên hệ nhóm mình qua email hoặc facebook**

[illegible]