TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN LẬP TRÌNH TÍNH TOÁN

TÊN ĐỀ TÀI: TÌM CHUỖI DNA VỚI PHƯƠNG PHÁP MỜ (FUZZY) SỬ DỤNG BIẾN ĐỔI FOURIER NHANH (FFT)

Người hướng dẫn**: TS.PHẠM MINH TUẤN**
Nhóm sinh viên thực hiện**:**

TÊN : HỒ BÁ THANH LỚP: 19TCLC_Nhật1 NHÓM:3
TÊN : NGUYỄN MINH QUANG LỚP: 19TCLC_Nhật1 NHÓM:3

Đà Nẵng, 12/2020

MỤC LỤC

DANH MỤC HÌNH VẼ

LỜI MỞ ĐẦU

Ngày nay Công nghệ thông tin đã phát triển với tốc độ nhanh chóng và ngày càng có nhiều triển vọng để phát triển mạnh hơn nữa. Công nghệ tin học đã được ứng dụng trong nhiều lĩnh vực như nghiên cứu khoa học, phát triển kinh tế, quân sự và trong nhiều loại hình nghệ thuật khác nhau.

Tất cả các nước đều đang cố gắng làm chủ kiến thức và tìm cách áp dụng thành tựu của Công nghệ thông tin vào mọi ngành kinh tế – xã hội của nhà nước.. Nghiên cứu công nghệ di truyền là một trong những lĩnh vực phát triển rất nhanh của ngành Công nghệ thông tin. Nó được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học và công nghệ, chẳng hạn như y học, kiến trúc, giải trí.... Nhiều nghiên cứu đã vượt ra những mong đợi của con người ,đặc biệt chúng ta đã tiến rất bộ vượt bậc trong nghành nghiên cứu về di truyền biến dị ở đây là về mặt sinh học,cấu trúc DNA,…

Một trong những lĩnh vực của nghiên cứu di truyền là tìm những đoạn DNA xấu,đột biến dị mang bệnh trong một chuỗi DNA để từ đó tìm ra được những cách khắc phục tốt nhất cho đoạn DNA đó. Đồ án giải quyết bài toán "Tìm chuỗi DNA bằng phương pháp mờ (Fuzzy) sử dụng phương pháp Fast Fourier Transform (FFT)".Qua các phương pháp nghiên cứu như tham khảo tài liệu,nghiên cứa cá nhân,làm việc nhóm,..Đồ án tìm hiểu và nghiên cứu trong phạm vi chuỗi cho trước để so sánh các thuật toán tìm kiếm với nhau,từ đó đưa ra các ưu nhược điểm của từng thuật toán.

Mặc dù đã hết sức cố gắng, nhưng do điều kiện thời gian và khả năng còn nhiều hạn chế nên đồ án vẫn đang còn nhiều thiếu sót. Rất mong nhận được sự góp ý, phê bình, đánh giá để nội dung của đồ án hoàn thiện hơn.

Đặc biệt chúng em xin bày tỏ lòng biết ơn sâu sắc tới TS. Phạm Minh Tuấn người đã trực tiếp hướng dẫn, định hướng,giúp đỡ để chúng em hoàn thành đồ án này.

Đà Nẵng, ngày 26 tháng 12 năm 2020

BẢNG PHÂN CÔNG NHIỆM VỤ

Điểm	**Phân Công Nhiệm Vụ**		**Chữ Ký của SV**
	Hồ Bá Thanh	-Đọc tài liệu,cài đặt và viết báo cáo về thuật toán Vét Cạn -Đọc tài liệu,viết báo cáo về phương pháp Fuzzy,DFT&IDFT -Tổng hợp làm báo cáo,Kết quả thực nghiệm,Làm Slide	
	Nguyễn Minh Quang	-Đọc tài liệu,cài đặt viết báo cáo về thuật toán FFT,phương pháp hàm sinh,nhân nhanh 2 đa thức -Hỗ trợ làm Slide	

1. TỔNG QUAN ĐỀ TÀI

Ngày nay cùng với sự phát tiển của công nghệ thì tương tác giữa con người và máy tính ngày càng dễ dàng hơn, đặc biệt là các thuật toán .Nó giúp con người tính toán một khối lượng công việc khổng lồ tối ưu hóa thời gian cũng như giảm bớt sự sai sót không đáng có. Đề tài của nhóm hướng tới việc tìm kiếm chuỗi DNA bằng phương pháp mờ (Fuzzy) sử dụng biến đổi Fourier nhanh(FFT) .Ở đây là tìm kiếm chuỗi T trong chuỗi S với độ đột biến k là khoảng cách giữa 2 chuỗi gần nhất. Ta có thể dùng thuật toán tìm kiếm chuỗi thông thường hiện nay(Vét Cạn) nhưng với độ phức tạp lớn thì thuật toán Vét Cạn sẽ bị hạn chế,từ đó nhóm xây dựng thuật toán với phương tìm kiếm nhanh (FFT) nhằm khắc phục hạn chế trên. Rút ra đánh giá xem mức độ tối ưu của các phương pháp tìm kiếm trên là như thế nào và đưa ra kết luận.

2. CƠ SỞ LÝ THUYẾT

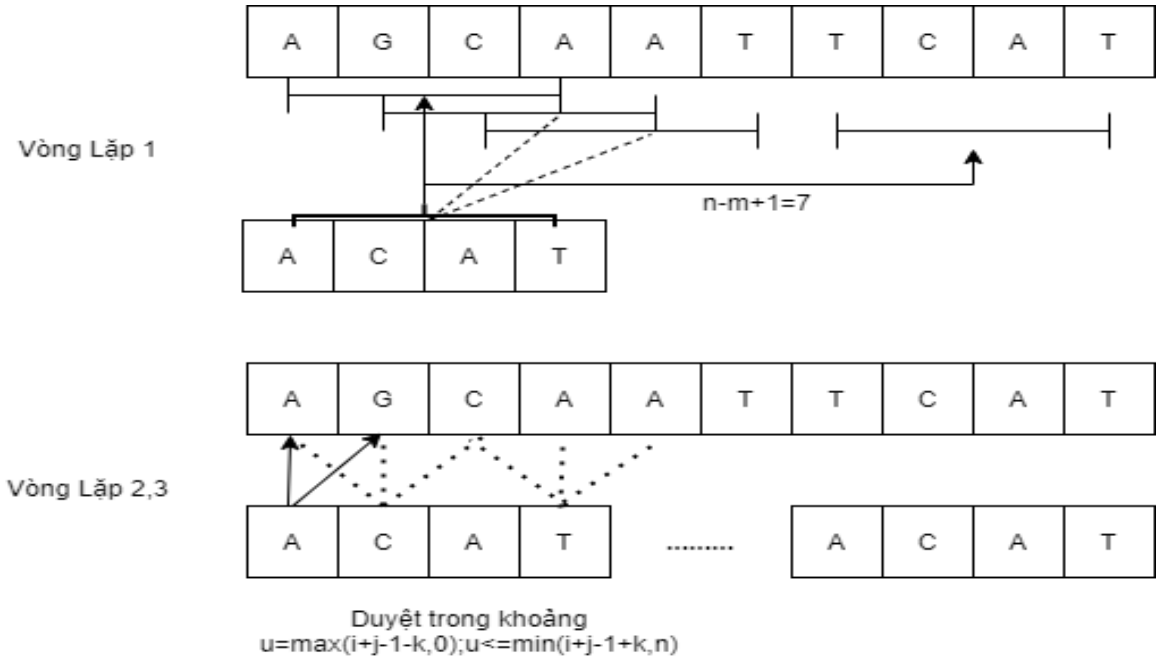
2.1 Ý tưởng

a)Thuật toán dùng phương pháp vét cạn

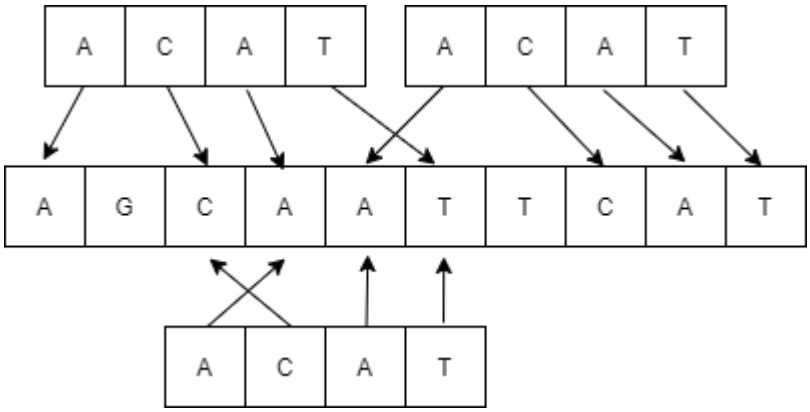
Dùng 3 vòng lặp để lần lượt so sánh các ký tự của chuỗi T so với chuỗi S.Trong khoảng cho phép k.

- Vòng lặp 1 để duyệt so khớp T với S.
- Vòng lặp 2 để chạy các ký tự của T.
- Vòng lặp 3 để so sánh trong khoảng ứng với k

Ví dụ: n=10;m=4;k=1; chuỗi S= AGCAATTCAT; chuỗi T=ACAT



Hình 1: Ví dụ các bước cho ý tưởng thuật toán Vét cạn



Vậy kết quả sau khi sử dụng thuật toán là 3.

➔ Vì chương trình dùng thuật toán duyệt theo tuần từ cho nên độ chính xác đã được kiểm nghiệm .Nhưng cùng vì duyệt tuần tự cho nên trong các bài có độ thuật toán cao sẽ không ứng dụng được.

b)Thuật toán dùng phương pháp FFT

Với mỗi kí tự c=A,G,C,T (hay là mỗi nucleobase),$f_c[N], g_c[N]$ là các mảng

- Ta quy ước với chuỗi S:

$f_c[i] = 1$ nếu tại vị trí i của chuỗi S có thể đặt kí tự c vào đó (kí tự c có thể đặt được vào chuỗi S nếu vị trị đó cách với kí tự c gần đó không quá k).

Ví dụ: AACCCG với k=2

$\rightarrow f_A[0] = 1, f_A[1] = 1, f_A[2] = 1, f_A[3] = 1, f_A[4] = 0, f_A[5] = 0$

- Ta quy ước với chuỗi T:

$g_c[i] = 1$ nếu tại vị trí length(T)-i-1 của chuỗi T bằng kí tự c

Ví dụ: TGCCC

$g_c[0] = 1$ (vì T[4]=”C”),tương tự ta có $g_c[1] = 1, g_c[2] = 1, g_c[3] = 0, g_c[4] = 0$

- Từ 2 mảng $f_c[i]$, $g_c[i]$đó ta sinh thành 2 đa thức :

$$P_c[x] = f_c[0] + f_c[1]x + \cdots + f_c[\text{length}(S) - 1]x^{(\text{length}(S)-1)}$$

$$Q_c[x] = g_c[0] + g_c[1]x + \cdots + g_c[\text{length}(T) - 1]x^{(\text{length}(T)-1)}$$

- Ta tiến hành nhân 2 đa thức này và thu được:

$G_c[x] = P_c[x].Q_c[x] = res_c[0] + res_c[1]x + \cdots + res_c[length(S) + length(T) - 2]x^{(length(S)+length(T)-2)}$. Ở đây ta sẽ áp dụng FFT trong việc nhân 2 đa thức này để cải thiện tốc độ thực hiện của thuật toán.

- Lúc đó theo mệnh đề của phương pháp sinh thì:

$$res_c[j] = f_c[0]g_c[j] + f_c[1]g_c[j-1] + \cdots + f_c[j]g_c[0]$$

→ $res_C[j]$là biểu diễn của số kí tự c có thể khớp được của hai chuỗi S và T bắt đầu từ vị trí j-|T|+1

Giả sử nếu tại vị trí j-|T|+1 thì 2 chuỗi có thể khớp được với nhau thì ta sẽ chứng minh lúc đó $res_C[j]$ bằng số kí tự c trong chuỗi T

Thật vậy :
Trong mảng $g_c[i]$ có số số 1 = số kí tự c trong chuỗi T

Giả sử $f_c[j-i]$= 0 và $g_c[i] = 1$ => $f_c[j-i]g_c[i]$= 0 lúc đó sẽ có ít nhất 1 kí tự trong chuỗi T không khớp được với chuỗi S khi đặt chuỗi T tại vị trí j-length(T)+1 của chuỗi S .

Vậy nếu $res_A[j] + res_G[j] + res_C[j] + res_T[j] = length(T)$ thì lúc đó chuỗi T có thể đặt tại vị trí j-|T|+1 để có thể khớp được với chuỗi S .

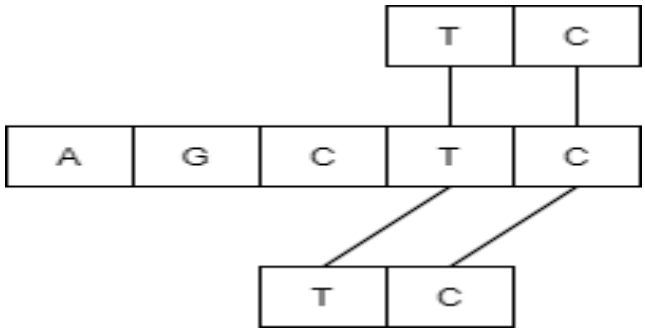
Ví dụ: S = AGCTC , T = TC,n=5,m=2,k=1

Xét kí tự A :
Ta có $f_A[0] = 1, f_A[1] = 1, f_A[2] = 0, f_A[3] = 0, f_A[4] = 0$
Và $g_A[0] = 0, g_A[1] = 0$
Suy ra $P_A(x) = 1 + x, Q_A(x) = 0 => G_A(x) = 0$. Rõ ràng vì không có kí tự A nào của chuỗi T khớp được với chuỗi S .
Xét kí tự C :
Ta có $f_C[0] = 0, f_C[1] = 1, f_C[2] = 1, f_C[3] = 1, f_C[4] = 1$
Và $g_C[0] = 1, g_C[1] = 0$
Suy ra $P_C(x) = x + x^2 + x^3 + x^4, Q_C(x) = 1 => G_C(x) = x + x^2 + x^3 + x^4$
Xét kí tự G :
Ta có $f_G[0] = 1, f_G[1] = 1, f_G[2] = 1, f_G[3] = 0, f_G[4] = 0$
Và $g_G[0] = 0, g_G[1] = 0$
Suy ra $P_G(x) = 1 + x + x^2, Q_G(x) = 0 => G_G(x) = 0$
Xét kí tự T :
Ta có $f_T[0] = 0, f_T[1] = 0, f_T[2] = 1, f_T[3] = 1, f_T[4] = 1$
Và $g_T[0] = 0, g_G[1] = 1$
Suy ra $P_G(x) = x^2 + x^3 + x^4, Q_G(x) = x => G_G(x) = x^3 + x^4 + x^5$
Ta có :

$$G_A(x) + G_C(x) + G_T(x) + G_G(x) = x + x^2 + 2x^3 + 2x^4 + x^5$$

Do hệ số của x^3, x^4 bằng độ dài của chuỗi T ,tức là có thể đặt chuỗi T vào vị trí $i = 3 - 2 + 1 = 2, i = 4 - 2 + 1 = 3$ để khớp với chuỗi S .

Thật vậy :



2.2 Cơ sở lý thuyết

a) Giới thiệu chung về DNA

Việc phát hiện ra DNA là một bước ngoặt rất lớn trong khoa học sinh học nói riêng và cuộc sống con người nói chung. DNA giúp con người giải quyết hầu hết những câu hỏi chúng ta đặt ra về ở các loài,các căn bệnh ,dị tật di truyền,khoa học quân sự,..và ngày càng phát hiện thêm rất nhiều cấu trúc DNA mới ,điều đó đã và đang tạo ra một khối lượng khổng lồ dữ liệu các chuỗi gen phục vụ cho y học hiện đại. Kích thước dữ liệu ngày càng tăng đặt ra vấn đề về chi phí không gian lưu trữ ,cũng như tìm kiếm phân tích chuỗi trong một đoạn DNA nào đó.

DNA(deoxyribonucleic acid) là phân tử mang thông tin di truyền quy định mọi hoạt động sống (sinh trưởng, sinh sản, phát triển v.v.) của các sinh vật và hầu hết virus. Thành phần của DNA bao gồm các đơn phân nucleotide. Mỗi nucleotide được cấu tạo từ một trong bốn loại nucleobase chứa nitơ hoặc là cytosine (C), guanine (G), adenine (A), hay thymine (T).[1]

Dạng đơn giản nhất của một DNA trong một tế bào là cấu trúc dây xoắn đôi, trong đó 2 sợi DNA đơn xoắn quanh nhau theo hình xoắn ốc thuận tay phải.Bộ gen của con người gồm khoảng 3 tỷ đặc trưng trên 23 cặp nhiễm sắc thể (NST). Do đó ,cơ sở dữ liệu gen là vô cùng lớn và phức tạp. Để lưu trữ,truy cập và xử lý dữ liệu này một cách hiệu quả là một nhiệm vụ vô cùng khó khăn. DNA có chức năng chính là lưu trữ ,bảo quản và truyền đạt thông tin di truyền về cấu trúc và toàn bộ các loại protein của cơ thể sinh vật.[1]

b) Các phương pháp tìm kiếm

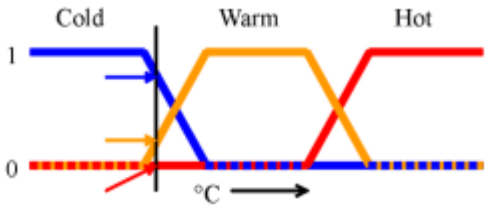
Đặc trưng phức tạp của một chuỗi DNA nằm ở chỗ đó là một chuỗi các chỉ số độ dài khác nhau biểu diễn một phạm vi có thể dự đoán được của các thành phần cơ bản cấu tạo nên DNA .Những đặc trưng phức tạp này cho phép tìm kiếm những cấu trúc lặp bên trong một nhiễm sắc thể(NST) hoặc qua nhiều NST .Và cũng chính những đặc trưng này được sử dụng để tìm ra khoảng cách tiến hóa và cấu trúc nên cây phát sinh loài.Nên khó có thể đưa ra 1 thuật toán tối ưu để tìm kiếm dành cho chuỗi DNA đã được phát triển từ 30 năm trước.

Hiện nay, kỹ thuật tìm kiếm dữ liệu chuỗi DNA được sử dụng rộng rãi trong sinh học. Để phân tích một chuỗi DNA ký hiệu là S, người ta tìm tất cả các lần xuất hiện của chuỗi T trong một chuỗi S. Bình thường chúng ta có thể sử dụng các thuật toán tìm kiếm chuỗi như Vét Cạn để giải quyết bài toán này. Tuy nhiên, thực tế là đoạn chuỗi DNA có thể chứa các đột biến nhỏ khiến việc tìm kiếm trở nên phức tạp. Chính vì vậy, các nhà khoa học đã đề xuất một định nghĩa mờ (fuzzy) để phù hợp với việc tìm kiếm đoạn DNA. Bây giờ chúng ta sẽ tìm hiểu kỹ hơn về các phương pháp này,ưu nhược điểm ở đâu .

c) Định nghĩa mờ Fuzzy

Lôgic mờ (Fuzzy logic) được phát triển từ lý thuyết tập mờ để thực hiện lập luận một cách xấp xỉ thay vì lập luận chính xác theo lôgic vị từ cổ điển. Lôgic mờ có thể được coi là mặt ứng dụng của lý thuyết tập mờ để xử lý các giá trị trong thế giới thực cho các bài toán phức tạp (Klir 1997).[2]

Một ứng dụng cơ bản có thể có đặc điểm là các khoảng con của một biến liên tục.[2] *Ví dụ:* Một đo đạc nhiệt độ cho phanh (anti-lock brake) có thể có một vài hàm liên thuộc riêng biệt xác định các khoảng nhiệt độ cụ thể để điều khiển phanh một cách đúng đắn. Mỗi hàm ánh xạ cùng một số đo nhiệt độ tới một chân giá trị trong khoảng từ 0 đến 1. Sau đó các chân giá trị này có thể được dùng để quyết định các phanh nên được điều khiển như thế nào.



Hình 2: Ví dụ đo đạc nhiệt độ dùng định nghĩa mờ Fuzzy

Trong hình, cold (lạnh), warm (ấm), và hot (nóng) là các hàm ánh xạ một thang nhiệt độ. Một điểm trên thang nhiệt độ có 3 "chân giá trị" — mỗi hàm cho một giá trị. Đối với nhiệt độ cụ thể trong hình, 3 chân giá trị này có thể được giải nghĩa là 3 miêu tả sau về nhiệt độ này: "tương đối lạnh", "hơi hơi ấm", và "không nóng".

Người ta hay nhầm lẫn mức độ đúng với xác suất. Tuy nhiên, hai khái niệm này khác hẳn nhau; độ đúng đắn của lôgic mờ biểu diễn độ liên thuộc với các tập được định nghĩa không rõ ràng, chứ không phải khả năng xảy ra một biến cố hay điều kiện nào đó.[2]

➔ Đề tài của nhóm ở đây nói một cách rõ hơn là giúp nhà khoa học đưa ra quyết định việc tìm kiếm DNA là như thế nào.

d) Phương pháp Vét Cạn

Vét cạn(brute force) là một phương pháp giải toán trong tin học: tìm nghiệm của một bài toán bằng cách xem xét tất cả các phương án có thể. Đây là cách tiếp cận cơ bản nhất để giải các bài toán trong tin học, phương pháp này thường được nghĩ đến đầu tiên trong quá trình thiết kế thuật toán để giải một bài toán trong tin học.[3]

Ưu điểm của phương pháp này là luôn đảm bảo tìm ra nghiệm đúng, chính xác. Tuy nhiên, hạn chế của phương pháp này là thời gian thực thi lâu, độ phức tạp lớn. Do đó vét cạn thường chỉ phù hợp với các bài toán có kích thước nhỏ.

e) Phương pháp DFT&IDFT [4]

1) Khái Niệm

-Tính Fourier rời rạc (**Discrete Fourier Transform-DFT**)là xác định chuỗi N giá trị phức {X(k)} khi biết trước chuỗi {x(n)} chiều dài N.

DFT : $X(k)=\sum_{n=0}^{N-1} x(n)W_N^{kn}$ với $0 \le k \le N-1$, $W_N = e^{-j\frac{2\pi}{N}}$

-Tính Fourier rời rạc ngược (**Inverse Discrete Fourier Transform-IDFT**) thì ngược lại xác định chuỗi N giá trị thực {x(n)} khi biết trước chuỗi {X(k)} chiều dài N.

IDFT: $x(n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn}$ với $0 \le n \le N-1$, $W_N = e^{-j\frac{2\pi}{N}}$

➔ Giải thuật tính DFT vẫn áp dụng cho việc tính IDFT

- Cách tính trực tiếp :
  - N^2 phép nhân phức
  - N(N-1) phép cộng phức

➔ Độ phức tạp cách tính $O(N^2)$

2) Tính chất

a) Tuyến Tính

Nếu: $x_1(n)_N \xleftrightarrow{DFT} X_1(k)_N$ $x_2(n)_N \xleftrightarrow{DFT} X_2(k)_N$

Thì: $a_1x_1(n)_N+a_2x_2(n)_N \xleftrightarrow{DFT} a_1X_1(k)_N + a_2X_2(k)_N$

b) Dịch Vòng

Nếu: $x(n)_N \xleftrightarrow{DFT} X(k)_N$

Thì: $x(n-n_0)_N \xleftrightarrow{DFT} W_N^{kn_0}X(k)_N$

c) Chập Vòng

Nếu : $x_1(n)_N \xleftrightarrow{DFT} X_1(k)_N$ $x_2(n)_N \xleftrightarrow{DFT} X_2(k)_N$

Thì: $x_1(n)_N * x_2(n)_N \xleftrightarrow{DFT} X_1(k)_N . X_2(k)_N$ (* ở đây là tổng chập)

f) Thuật toán FFT cơ số 2 [4]

***1)* Khái niệm**

Từ cách tính DFT,IDFT như trên với độ phức tạp lớn như vậy gây mất thời gian cũng như khó khả khi áp dụng cho nên FFT ra đời để giảm độ phức tạp thuận toán xuống còn O(Nlog(N)).

***2)* Công thức thuật toán**

- Giả Thiết dãy x(n) có độ dài N= 2^M,nếu không có dạng lũy thừa 2 thì thêm vài mẫu 0 vào dãy x(n):

→ $X(k)= \sum_{n=0}^{N-1} x(n) W_N^{kn}$ Với: k=0,1,....,N-1

$= \sum_{n\ even} x(n)\, W_N^{kn} + \sum_{n\ old} x(n)\, W_N^{kn}$

$= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2km} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)}$

Đặt: $X_0(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_{N/2}^{km}$, $X_1(k) = \sum_{m=0}^{(N/2)-1} x(2m+1) W_{N/2}^{km}$

→ $X(k)= X_0(k) + W_N^k X_1(k)$ Với: k=0,1,...,N – 1

- $X_0(k)$ là DFT của N/2 điểm ứng với chỉ số n chẵn với k=0,1,..,N/2
- $X_1(k)$ là DFT của N/2 điểm ứng với chỉ số n lẽ với k=0,1,..,N/2

Chú ý rằng $X(k) = X_0(k) + W_N^k X_1(k)$ với $k = 0, \dots, \frac{N}{2} - 1$

Và $X\left(k + \frac{N}{2}\right) = X_0(k) - W_N^k X_1(k), k = 0,.., \frac{N}{2} - 1$ (do $W_N^k = -W_N^{k+\frac{N}{2}}$)

- Sau đó đánh lại chỉ số theo thứ tự các mẫu x(n) ,tiếp tục phân chia DFT của N/2 điểm thành 2 DFT của N/4 điểm theo chỉ số n chẵn và lẻ và cứ thế tiếp tục phân chia cho đến khi nào còn DFT 2 điểm thì dừng lại.

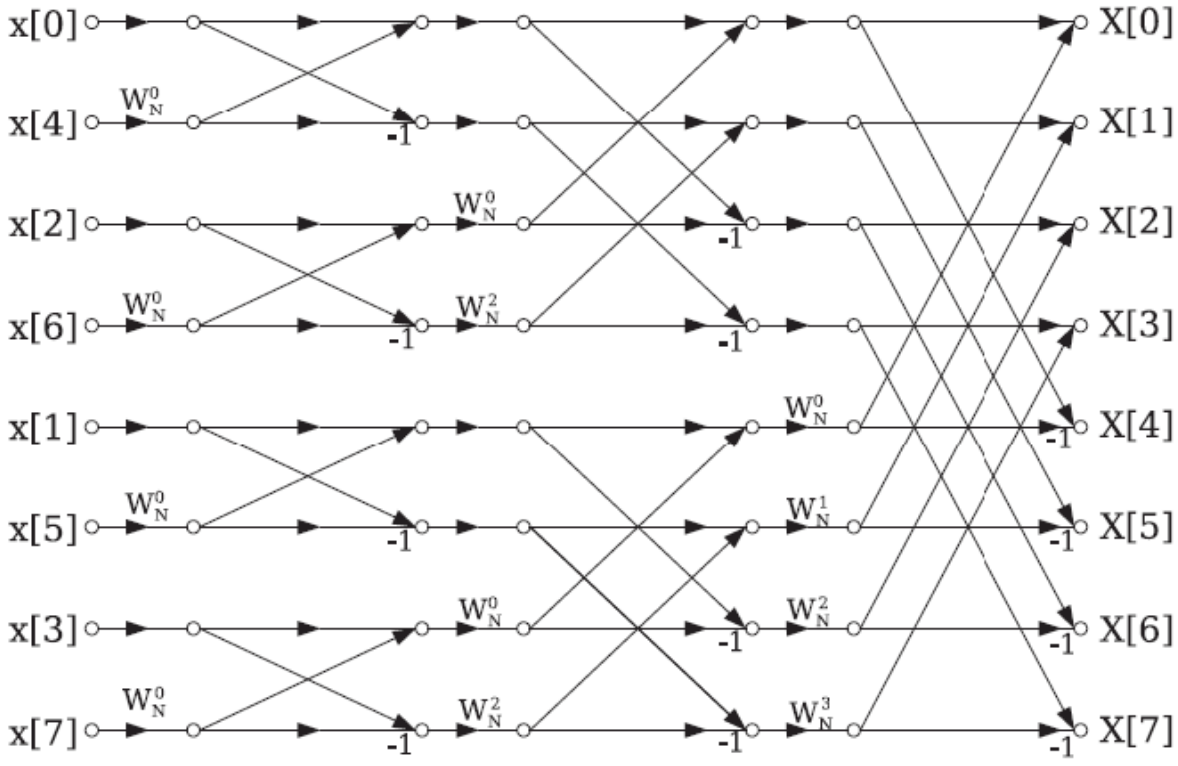
- Ví dụ $X_0(k)$ được phân chia sau:

$X_0(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_{N/2}^{km} = \sum_{m=0}^{(N/2)-1} g(m) W_{N/2}^{km}$

$= \sum_{m=0,2,4,..}^{(N/2)-1} g(m) W_{N/2}^{km} + \sum_{m=1,3,5,..}^{(N/2)-1} g(r) W_{N/2}^{km}$

$= \sum_{l=0}^{(N/4)-1} g(2l) W_{N/4}^{km} + W_{N/2}^{k} \sum_{l=0}^{(N/4)-1} g(2l+1) W_{N/4}^{km}$

$= X_{00}(k) + W_{N/2}^{k} . X_{01}(k)$



Hình 3: Ví dụ biểu diễn cho thuật toán với n=8

g) Phương pháp hàm sinh [5]

1) Lý thuyết

Phương pháp hàm sinh là một phương pháp dùng trong bài toán đếm tổ hợp . Ý tưởng của phương pháp này xuất phát từ việc đếm hệ số x_r trong khai triển hàm sinh với r là số phần tử được chọn từ n đối tượng mà đề bài cho .

Định Nghĩa:

Hàm sinh của dãy số thực $a_0, a_1, \dots, a_n, .,$ được xác định bởi

$$P(x) = a_0 + a_1x + \cdots . + a_nx^n + \cdots.$$

Mệnh Đề (*Công thức xác định hệ số tích của hai hàm sinh*):

Cho hai hàm sinh của dãy $(a_n), (b_n)$ lần lượt là :

$$A(x) = a_0 + a_1x + \cdots + a_nx^n + \cdots$$
$$B(x) = b_0 + b_1x + \cdots + b_nx^n + \cdots$$

Đặt G(x)=A(x)B(x)

Khi đó hệ số của x^r trong G(x) là :

$$a_0b_r + a_1b_{r-1} + \cdots + a_rb_0$$

2) Một số ứng dụng của phương pháp hàm sinh trong đếm tổ hợp

Ví dụ 1: Vào ngày nghỉ chủ nhật, cô Hoa đi chơi và mua quà là 12 quả cam cho 3 đứa trẻ An, Bình, Chi.Hỏi cô Hoa có bao nhiêu cách phân phối 12 quả cam sao cho An có

ít nhất 4 quả, Bình và Chi mỗi người đều có ít nhất 2 quả, nhưng Chi không được nhiều hơn 5 quả?

Lời giải:
Hàm sinh của số cách chọn quả cho An là :

$$A(x) = x^4 + x^5 + .. + x^8$$

Hàm sinh của số cách chọn quả cho Bình và Chi lần lượt là :

$$B(x) = x^2 + x^3 + \cdots + x^6$$
$$C(x) = x^2 + x^3 + .. + x^5$$

Ta có: G(x)=A(x)B(x)C(x)
Do ta chỉ cần tìm hệ số của 12 trong khai triển của G(x)
Mà ta có hệ số của 12 của G(x) sau khi khai triển là 14
Do đó có 14 cách để cô Hoa chia cho 3 bạn An,Bình , Chi

***Ví dụ 2**:* Có bao nhiêu cách chọn ra 15 USD từ 20 người nếu 19 người đầu,một người có thể đưa ra nhiều nhất 1 USD ,người thứ 20 có thể đưa ra 1 USD hoặc 5 USD hoặc không USD nào.

Lời giải:

Hàm sinh cho số cách chọn nhiều nhất 1 USD từ 19 người là:

$$A(x) = (1 + x)^{19}$$

Hàm sinh cho số cách chọn 1 USD hoặc 5 USD hoặc không USD nào ở người thứ 20:

$$B(x) = 1 + x + x^5$$

Hàm sinh cho số cách chọn ra 15 USD là:

$$G(x) = A(x)G(x) = (1 + x)^{19}(1 + x + x^5)$$

Chúng ta tìm hệ số của x^{15} trong khai triển của G(x),Ta có:

$$(1 + x)^{19} = \sum_{k=0}^{19} C_{19}^{k}\, x^{19-k}$$

Đặt a_r là hệ sô của x^r trong khai triển A(x) , b_r là hệ sô của x^r trong khai triển B(x)

Khi đó ta có: $a_r = C_{19}^{r}$ và $b_0 = b_1 = b_5 = 1$

Vậy hệ số của x^{15} trong khai triển của G(x) là:

$$a_{15}b_0 + a_{14}b_1 + a_{13}b_2 + \cdots + a_0b_{15}$$

Ta có:

$$a_{15}b_0 + a_{14}b_1 + a_{10}b_5 = C_{19}^{15}+C_{19}^{14} + C_{19}^{10} = 107882$$

Vậy có 107882 cách chọn ra 15 USD thỏa mãn điều kiện bài toán.

h) Thuật toán nhân hai đa thức bằng FFT [6]

Đầu tiên ta có biến đổi DFT của X(k) có dạng đa thức . Thật vậy :

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{kn} = \sum_{n=0}^{N-1} x(n)\left(W_N^k\right)^n = \sum_{n=0}^{N-1} x(n)a^n \ , a = W_N^k$$

Do đó phép biến đổi DFT của một đa thức A(x) có các hệ số $a_0, a_2, .., a_{n-}$、được định nghĩa như sau :

$$DFT(a_0, a_1, .., a_{n-1}) = \left(X(w_n^0), X(w_n^1), \dots, X(w_n^{n-1})\right) = (X(0), X(1), .., X(n-1))$$

-Ta có xét 2 đa thức $A(x), B(x)$ có bậc lần lượt là n, m:

$$A(x) = a_0 + a_1x + \cdots + a_nx^n + a_{n+1}x^{n+1}+..+a_{n+m}x^{n+m}$$
$$B(x) = b_0 + b_1x + \cdots + b_mx^m + b_{m+1}x^{m+1}+..+a_{n+m}x^{n+m}$$

Với a_i, b_i là hệ số của đa thức $A(x), B(x)$ trong đó $a_i = 0$ với $i > n$; $b_i = 0$ với $i > m$

Lưu ý rằng nếu $n + m + 1$ không phải là lũy thừa của 2 thì đơn giản ta chỉ cần thêm a_ix^i cho đến khi i có dạng lũy thừa của 2 ,trong đó $a_i = 0$ khi ta thêm vào .

-Lúc đó ta sẽ tính nhanh đa thức $C(x) = A(x)B(x)$.

Chú ý rằng hệ số của x^r trong đa thức C(x) là :

$$c_r = a_0b_r + a_1b_{r-1} + \cdots + a_rb_0 \ (*)$$

(*) tương đương với việc tính tổng chặp của 2 dãy :

$$A[n] = \{a_0, a_1, .., a_n, .., a_{n+m}\}$$
$$B[n] = \{b_0, b_1, \dots, b_n, b_{n+m}\}$$

Và $C[n] = A[n] * B[n], C[n] = \{c_0, c_1, .., c_{n+m}\}$ (**)

Từ (**) cộng với tính chất của biến đổi DFT ta thực hiện tính nhanh $C[n]$ như sau:

+)Biến đổi DFT của 2 dãy $A[n], B[n]$ ta thu được $DFT(A[n]), DFT(B[n])$

Từ đó ta có: $DFT(C[n]) = DFT(A[n]).DFT(B[n])$ (tính chất chặp vòng của DFT)

+)Sau đó ta thu được hệ số của dãy $C[n]$ bằng cách :

$$C[n] = A[n] * B[n] = InverseDFT(DFT(A[n]).DFT(B[n])))$$

Từ đó suy ra hệ số của $C[n]$ và đó cũng là hệ số của đa thức $C(x)$.

(Chú ý rằng ở đây chúng ta sẽ dùng FFT để biến đổi DFT của các dãy $A[n], B[n], C[n]$ để giúp thuật toán của ta hoạt động tốt hơn)

Ví dụ : Cho hai đa thức $A(x) = 1 + 2x + 5x^2, B(x) = 1 + 2x$.

Ta viết lại thành : $A(x) = 1 + 2x + 5x^2 + 0.x^3, B(x) = 1 + 2x + 0x^2 + 0x^3$

Ta có $A[3] = \{1,2,5,0\}, B[3] = \{1,2,0,0\}.$

Ta biến đổi DFT của A và B và thu được :

$$DFT(A[3]) = \{8, -4 - 2i, 4, -4 + 2i\}$$

$$DFT(B[3]) = \{3,1 - 2i, -1,1 + 2i\}$$

Ta có $DFT(A[3])DFT(B[3]) = \{24, -8 + 6i, -4, -8 - 6i\}$

Suy ra $C[n] = IDFT\big(DFT(A[3])DFT(B[3])\big) = \{1,4,9,10\}$

Vậy $C(x) = A(x)B(x) = 1 + 4x + 9x^2 + 10x^3$

3. TỔ CHỨC CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN

3.1 Phát biểu bài toán

DNA(deoxyribonucleic acid) là phân tử mang thông tin di truyền quy định mọi hoạt động sống (sinh trưởng, sinh sản, phát triển v.v.) của các sinh vật và hầu hết virus. Thành phần của DNA bao gồm các đơn phân nucleotide. Mỗi nucleotide được cấu tạo từ một trong bốn loại nucleobase chứa nitơ—hoặc là cytosine (C), guanine (G), adenine (A), hay thymine (T). Để phân tích mội chuỗi DNA ký hiệu là S, người ta tìm tất cả các lần xuất hiện của chuỗi T trong một chuỗi S. Bình thường chúng ta có thể sử dụng các thuật toán tìm kiếm chuỗi như Vét Cạn để giải quyết bài toán này. Tuy nhiên, thực tế là đoạn chuỗi DNA có thể chứa các đột biến nhỏ khiến việc tìm kiếm trở nên phức tạp. Chính vì vậy, các nhà khoa học đã đề xuất một định nghĩa mờ (fuzzy) để phù hợp với việc tìm kiếm đoạn DNA có các đột biến như sau:

− Chuỗi T tồn tại trong chuỗi S ở vị trí i ($1 \le i \le |S| - |T| + 1$), nếu sau khi đặt chuỗi T cùng với vị trí này, mỗi ký tự của chuỗi T tương ứng với một số ký tự của cùng một giá trị trong chuỗi S cách nhau nhiều nhất là số k cho trước. Nếu k = 0 thì việc tìm kiếm như tìm kiếm thông thường.

Input:

▪ Dòng đầu tiên là 3 số nguyên thể hiện chiều dài của chuỗi S, chuỗi T và số k.

($1 \le |T| \le |S| \le 200000$, $0 \le k \le 200000$)

▪ Dòng thứ hai chứa chuỗi S.

▪ Dòng thứ ba chứa chuỗi T.

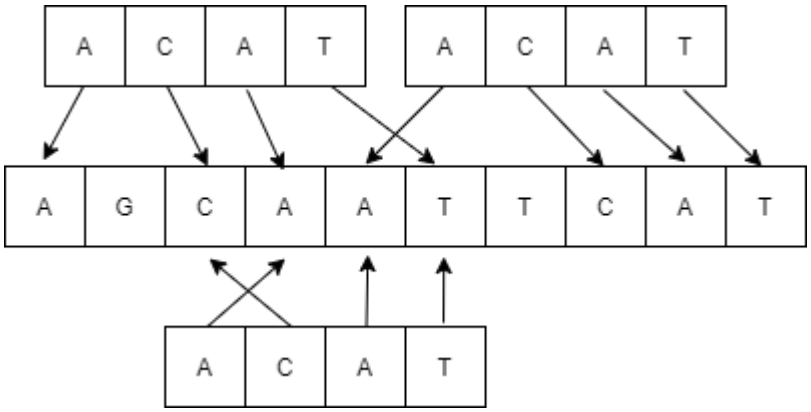
▪ Cả hai chuỗi chỉ bao gồm các chữ cái viết hoa ' A ', ' T ', ' G ' và ' C '.

Output:

▪ Số lần xuất hiện của chuỗi T trong một chuỗi S

Ví dụ: n=10, m=4, k=1, chuỗi S=AGCAATTCAT, chuỗi T = ACAT

Ta có thể đặt chuỗi T trong chuỗi S tại các vị trí như sau:



➔ Số lần có thể đặt chuỗi T vào chuỗi S là 3

3.2 Cấu trúc dữ liệu

a) Các thư viện sử trong bài

- <stdio.h>: Cung cấp cốt lõi của những khả năng nhập trong C. Tập tin này bao gồm họ hàm printf

- <string.h>: Để thao tác với loại dãy ký tự.

- <math.h>: Dùng cho việc tính các hàm số thông dụng.

b) Các biến quan trọng sử dụng trong chương trình:

- int N=600000 :hằng số để định nghĩa độ dài của mảng

- struct complex : Dùng để định nghĩa lại cấu trúc số phức

- const double pi=arcos(-1) : số pi
- char s[N],t[N] : chuỗi S và chuỗi T

- int n,m: độ dài lần lượt của chuỗi S và chuỗi T

- int k : khoảng cách lớn nhất của 2 kí tự giống nhau của 2 chuỗi S,T

- complex a[N], b[N]: mảng số phức sử dụng để biến đổi FFT và lưu giá trị mảng $f_c[N], g_c[N]$

3.3 Thuật toán

a) Thuật toán đảo BIT [6]

Trước khi cài đặt thuật toán FFT , chúng ta cần phải cài đặt thuật toán đảo Bit . Thuật toán đảo Bit dùng để có thể hoán đổi những phần tử có bit đảo ngược nhau như bảng dưới .

Thuật toán đảo BIT được cài đặt như sau :

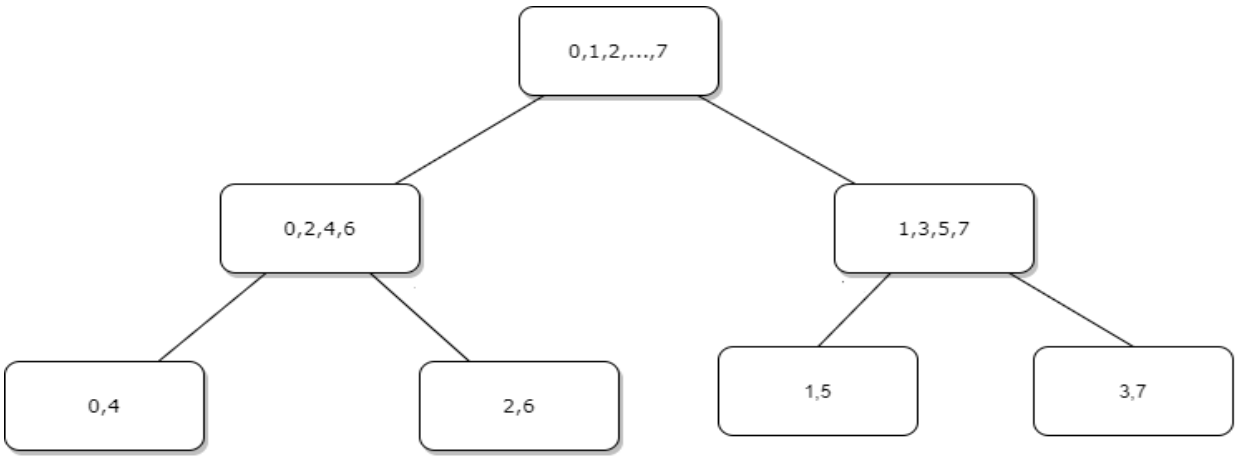
```
void reverseBIT(complex *a,int n)
{
   for (int i=1,j=n/2;i<n-1;i++)
        {
            if (i<j) swap(a[i],a[j]);
            int p=n/2;
            while (j>=p) j-=p,p/=2;
            j+=p;
        }
}
```

Bảng 1: Mô tả quy luật đảo bit với n=8

Chỉ số lúc chưa hoán đổi	Bit	Đảo bit	Chỉ số sau khi hoán đổi
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

➔ Với mỗi $i = 1 \rightarrow n - 2$ thì vòng while ở trong vòng for sẽ duyệt lần lượt các bit của j.Do vậy nên độ phức tạp của thuật toán này là $O(nlog_2(n))$

b) Thuật Toán FFT [6]



Hình 4: Ví dụ mối quan hệ giữa các phần tử trong biến đổi FFT và thứ tự với n=8

Ở đây ta có thể thấy rằng ta phải tính được biến đổi FFT của lần lượt các cặp phần tử (0,4) ; (2,6) ; (1,5) ; (3,7). Sau đó theo giải thuật FFT cơ số 2 với các biến đổi FFT của các cặp phần tử ta có thể tính được biến đổi FFT của 2 nhóm 4 cặp phần tử là (0,2,4,6) và (1,3,5,7) . Cuối cùng ta đi tính biến đổi FFT của 8 phần tử bằng 2 nhóm 4 cặp phần tử đó .

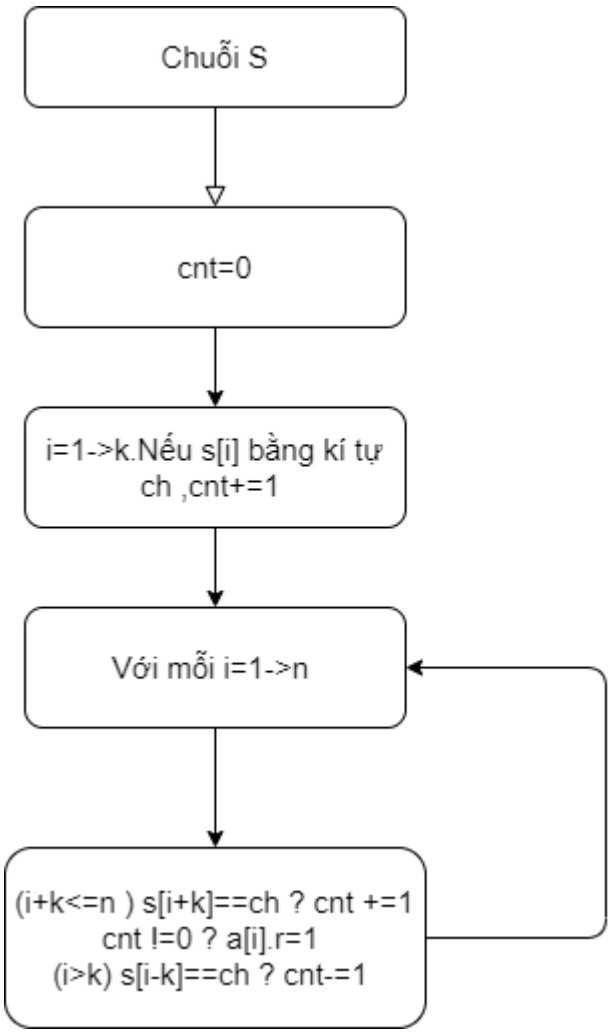
Sau đây là cách cài đặt thuật toán FFT :

```
void fft(complex *a,int n,int invert)
{
    reverseBIT(a,n);
    for (int j=2;j<=n;j*=2)
    {
        complex wn(cos(2*pi/j*invert),sin(2*pi/j*invert));
        for (int i=0;i<n;i+=j)
        {
            complex w(1,0);
            for (int k=i;k<i+j/2;k++)
            {
```

```
                    complex u=a[k],t=a[k+j/2]*w;
                    a[k]=u+t;a[k+j/2]=u-t;
                    w=w*wn;
                }
            }
        }
        if (invert==-1)
        for (int i=0;i<n;i++) a[i].r/=n;
}
```

➔ Với 3 vòng for ở trên thì độ phức tạp của thuật toán này là: $O(\log_2(n).\frac{n}{\log_2(n)}.\log_2(n))$ .Hay độ phức tạp được viết lại thành là $O(nlog_2(n))$

c) Thuật toán chuyển mỗi kí tự trong chuỗi cần được phân tích S và chuỗi cần tìm T sang dãy 0,1



Hình 5: Sơ đồ thuật toán chuyển kí tự trong chuỗi được phân tích sang dãy 0,1

- Với một chuỗi cần được phân tích S . a[N] là mảng số phức lưu khả năng có thể đặt được kí tự c (c=A,G,C,T) vào mỗi vị trí (kí tự c có thể đặt được vào chuỗi S nếu vị trị đó cách với kí tự c gần nhất không quá k),a[i].r=1 nếu đặt được còn không ngược lại là a[i].r=0 .
a[i].i=0 (a[i].i là phần ảo của a[i],a[i].r là phần thực của a[i])

- Với chuỗi cần tìm T. b[N] là mảng số phức lưu khả năng có thể đặt được kí tự c vào đó (kí tự c có thể đặt vào vị trí i nếu T[length(T)-i+1]=c) .

Sau đây là phần cài đặt thuật toán của 2 chuỗi S và T này:

```
void setStringtobinary(char ch)
```

```
{       int cnt=0 ;
        for (int i=1;i<=k;i++) cnt+=s[i]==ch;
        for (int i=1;i<=n;i++)
        {
        if (i+k<=n) cnt+=s[i+k]==ch;
        if (cnt) a[i].r=1;
        if (i>k) cnt-=s[i-k]==ch;
        }
        for (int i=1;i<=m;i++) b[i].r=t[m+1-i]==ch;
}
```

Ví dụ: Chuỗi S=AGCATTCA ,n=10,k=1
Ta sẽ xây dựng mảng $f_A[10] = a[i].r$ như sau :
Khởi tạo cnt=0 ,biến cnt ở đây dùng để lưu số kí tự A cách vị trí hiện tại không quá k.
Ban đầu ta duyệt từ 1 đến k ,nếu có kí tự là A thì cộng vào cnt thêm 1 .
Bây giờ với mỗi vị trí i ($1 \leq i \leq n$) :
Nếu kí tự thứ $i + k$ là kí tự A thì cnt sẽ cộng thêm vào 1 . Điều đó có nghĩa là mọi vị trí từ i đến $i + k$ đều có thể đặt kí tự A (vì cách kí tự A gần nhất không quá k)
Nếu cnt khác 0 thì $a[i].r = 1$
Nếu kí tự thứ $i - k$ là kí tự A thì cnt sẽ 1 . Điều đó có nghĩa là mọi vị trí từ $i + 1$ đến n cách kí tự A ở vị trí $i - k$ hơn k .
Với chuỗi S ở trên ,ta biến đổi $f_A[10]$ thành :
S : A G C A T T C A
f: 1 1 1 1 1 0 1 1

➔ Với chỉ một vòng for khi duyệt chuỗi S và T thì độ phức tạp của thuật toán : $O(n)$

d) Thuật toán nhân nhanh hai đa thức [6]

Với a[N],b[N] đã biết sau khi sử dụng thuật toán chuyển mỗi kí tự trong 2 dãy thành dãy 0,1 . Ta có thể cài đặt thuật toán nhân nhanh 2 đa thức với các hệ số a[N],b[N] như sau :

```
void fft_for_char(char ch)
{
int nn=1;
while (nn<=n+m) nn*=2;
for (int i=0;i<nn;i++) a[i].r=a[i].i=b[i].r=b[i].i=0;
setStringtobinary(ch);
fft(a,nn,1);fft(b,nn,1);
for (int i=0;i<nn;i++) a[i]=a[i]*b[i];
```

```
fft(a,nn,-1);
for (int i=0;i<nn;i++) res[i]+=(int)(a[i].r+0.1);
}
```

Ở đây sau khi nhân đa thức thì a[i] lưu giá trị số kí tự c có thể khớp tại vị trí i-length(T)+1

Sau mỗi lần nhân đa thức như thế ,ta cộng vào biến thêm vào res[i] với một a[i] .

Nếu như ở đó res[i]=length(T) thì chuỗi T có thể khớp với chuỗi S tại vị trí i-length(T)+1. ($res[i] = res_A[i] + res_G[i] + res_C[i] + res_T[i]$)

➔ Với lần gọi tính FFT của hai dãy số phức $a[N], b[N]$ với độ dài là $nn \sim 2(n+m)$thì độ phức tạp của thuật toán là: $O((n+m)\log(n+m))$

e) Thuật toán vét cạn(BruteForce)

Với chuỗi S và chuỗi T đầu vào, ta cài đặt thuật toán vét cạn cho bài này như sau:

```
void BruteForce(){
s=" "+s;
t=" "+t;
int cnt=0;
for( int i=1 ;i<=n-m+1;i++)
{   int ans=0;
    for( int j=1;j<=m;j++)
    {   char temp=t[j];
        for(int u=max(i+j-1-k,0);u<=min(i+j-1+k,n);u++)
        {   if(temp==s[u] && u >= 1 && u <= n)
            {
                ans+=1;
                break;
            }
        }
    }
```

```
        if(ans==m) cnt+=1;
    }
    cout<<cnt;
}
```

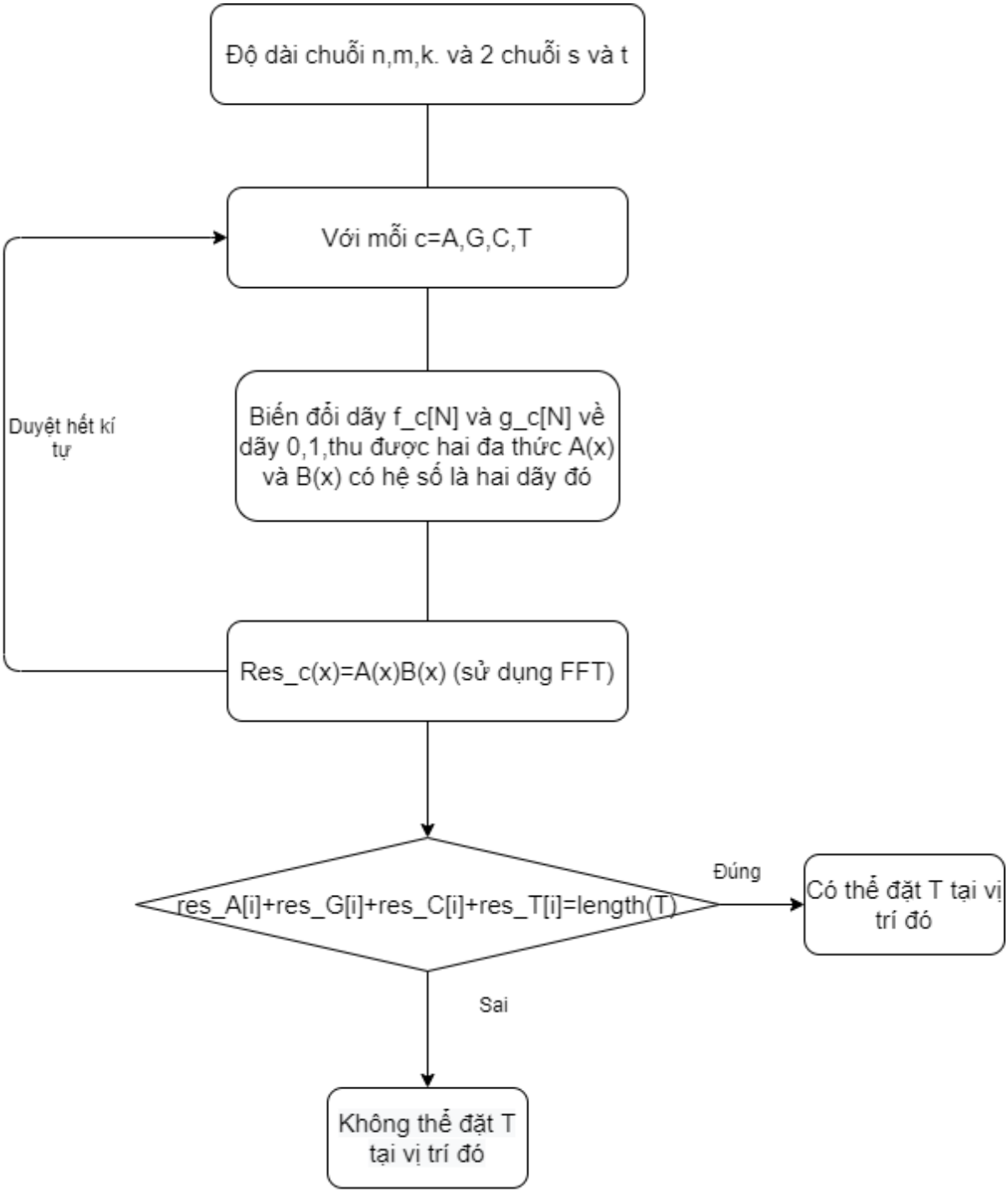
➔ Với 3 vòng for có độ dài lần lượt là $n - m + 1, m, 2k$ thì độ phức tạp của thuật toán sẽ là : $O((n - m + 1)m2k) \sim O(nmk)$

4. CHƯƠNG TRÌNH VÀ KẾT QUẢ

4.1 Tổ chức chương trình

a) Chương trình sử dụng phương pháp FFT

Sơ đồ thuật toán:



Hình 6: Sơ đồ thuật toán FFT tìm chuỗi

Các bước thực hiện chương trình:

- Ban đầu chúng ta tạo một struct số phức (complex) sau đó tạo các phép toán +,-,* cho số phức

- Khởi tạo các mảng số phức a[N] và b[N] để thực hiện FFT ,khởi tạo mảng char s[N],p[N] để lưu chuỗi ,khởi tạo mảng số nguyên res[N] để lưu số kí tự khớp

- reverseBIT(complex *a,int n) với tham số đầu vào là mảng số phức ,số lượng phần tử cần biến đổi là hàm dùng để hoán đổi những cặp phần tử mà chỉ số của 2 phần tử đó là đảo bit của nhau

- fft(complex *a,int n,int invert) có tham số đầu vào mảng số phức ,số lượng phần tử cần biến đổi . Hàm này dùng để tính FFT và IFFT của một mảng số phức cho trước , trong đó nếu invert=1 thì biến đổi FFT còn nếu invert=-1 thì là IFFT .Trong hàm FFT này chúng ta sẽ gọi hàm reverseBIT(complex *a,int n) để có thể hoán đổi những cặp phần tử mà chỉ số của 2 phần tử đó là đảo bit của nhau từ đó chúng ta có thể tính FFT của mảng số phức này .

- void setStringtobinary(char ch) với tham số đầu là một kí tự c(c='A','C','G','T') là hàm dùng để chuyển mỗi kí tự trong chuỗi cần được phân tích và chuỗi cần tìm sang dãy 0,1

- void multiplePoly(char ch) với tham số đầu vào là một kí tự c(c='A','C','G','T') là hàm dùng để nhân hai đa thức có hệ số đã tìm ở hàm setStringtobinary(char ch) cho mỗi kí tự

-void cactch_error() là hàm dùng để kiểm tra dữ liệu đầu vào để kiểm tra dữ liệu đầu vào có thỏa hay không

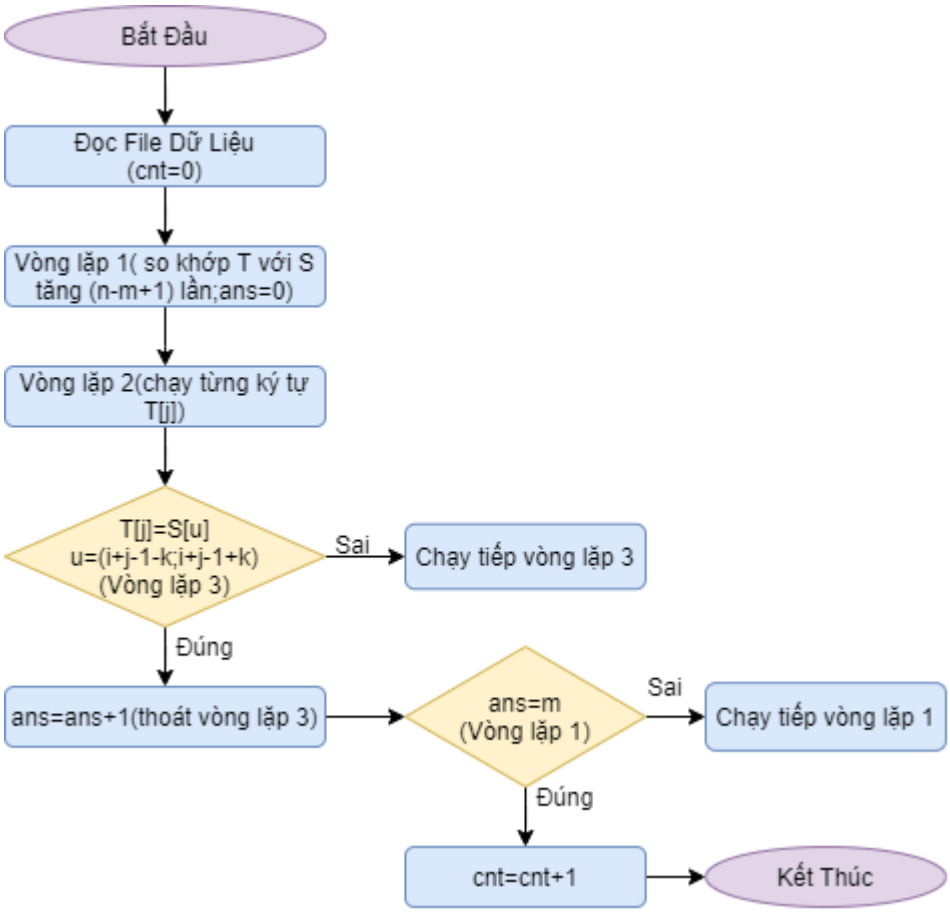
- void input() là hàm dùng để nhập dữ liệu đầu vào .

- void solve() là hàm dùng để tìm và in ra kết quả của bài toán

- int main() trong hàm này chỉ gồm 2 hàm input() và solve()

b) Chương trình sử dụng phương pháp vét cạn

Sơ đồ thuật toán:



Hình 7: Sơ đồ thuật toán Vét Cạn tìm kiếm chuỗi

Các bước thực hiện:

- Đầu tiên ta đọc file dữ liệu các chuỗi S, T trong file "input.txt"

- Sau lời gọi hàm Void BruteForce() với tham số đầu vào là mảng các ký tự chuỗi S,T số lượng chuỗi S,T là n và m cũng như khoảng cách cho phép ngắn nhất là k

- Vòng for đầu tiên của hàm là chạy so khớp của chuỗi T với S, tăng lên dần (n-m+1) lần

-Vòng for thứ 2 lấy từng ký tự của chuỗi T so sánh với khoảng cho phép(for thứ 3) k

-Nếu trong khoảng cho phép đáp ứng đủ điều kiện giống chuỗi S thì tăng biến đếm chuỗi T

-Qua các lần so sánh các ký tự của T so với S nếu biến đếm ans bằng độ dài chuỗi T thì ta được 1 lần so sánh hợp lệ.

c) Đánh giá độ phức tạp thuật toán

Thuật toán vét cạn có độ phức tạp là **O(nmk)**

Thuật toán FFT có độ phức tạp là **O((n+m)log(n+m))**

Với: n là độ dài chuỗi S

m là độ dài chuỗi T

k là khoảng cách 2 chuỗi gần nhau nhất

4.2 Ngôn ngữ cài đặt

Đồ án của nhóm cài đặt chương trình trên ngôn ngữ c++

4.3 Kết quả

4.3.1 Giao diện chính của chương trình

a) Giao diện chính của chương trình thuật toán Vét cạn

```
+-------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                               |
|                  TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)              |
|                                  Thuat Toan Vet Can (Brute Force)                                           |
+-------------------------------------------------------------------------------------- ----------------------+
|                                                                                                             |
|       THANH VIEN NHOM         : HO BA THANH                                                                 |
|                                 NGUYEN MINH QUANG                                                           |
|       LOP                     : 19TCLC_NHAT1                                                                |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                              |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
```

Hình 8: Giao diện chính của chương trình dùng phương pháp Vét cạn

b) Giao diện chính của chương trình thuật toán dùng FFT

- Ban đầu chương trình sẽ hiện lên tên đề tài , thành viên nhóm ,giáo viên hướng dẫn và menu chọn nhập dữ liệu đầu vào bằng console hoặc là file.

```
+-------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                               |
|                  TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)              |
+-------------------------------------------------------------------------------------------------------------+
|                                                                                                             |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                               |
|                             HO BA THANH                                                                     |
|       LOP                 : 19TCLC_NHAT1                                                                    |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                  |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
```

Hình 9: Giao diện chính của chương trình dùng phương pháp FFT

- Nếu nhập khác hai lựa chọn trên chương trình sẽ báo lỗi và bắt nhập lại đến khi nào đúng .

```
+-------------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                                   |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
+-------------------------------------------------------------------------------------------------------------------------+
|                                                                                                                         |
|       THANH VIEN NHOM : NGUYEN MINH QUANG                                                                               |
|                         HO BA THANH                                                                                     |
|       LOP       : 19TCLC_NHAT1                                                                                          |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                          |
|                                                                                                                         |
+-------------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
3
Lua chon khong hop le
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
```

Hình 10: Giao diện chính của chương trình khi lỗi lựa chọn

- Nếu nhập 1 thì chương trình sẽ yêu cầu nhập từ console . Sau đó chương trình sẽ in ra kết quả với dữ liệu nhập vào .

```
+-------------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                                   |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
+-------------------------------------------------------------------------------------------------------------------------+
|                                                                                                                         |
|       THANH VIEN NHOM : NGUYEN MINH QUANG                                                                               |
|                         HO BA THANH                                                                                     |
|       LOP       : 19TCLC_NHAT1                                                                                          |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                          |
|                                                                                                                         |
+-------------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
1
n=10
m=4
k=1
Chuoi S :AGCAATTCAT
Chuoi T :ACAT
Dau vao hop le
Chuoi S : AGCAATTCAT
Chuoi T : ACAT
n=10,m=4,K =1
So lan xuat hien chuoi T trong chuoi S la :3
```

Hình 11: Giao diện chính của chương trình khi nhập console đúng

- Nếu nhập 2 thì chương trình sẽ đọc file input.txt . Sau đó chương trình sẽ in ra màn hình dữ liệu nhập vào và kết quả nhận được

```
+-------------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                                   |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
+-------------------------------------------------------------------------------------------------------------------------+
|                                                                                                                         |
|       THANH VIEN NHOM : NGUYEN MINH QUANG                                                                               |
|                         HO BA THANH                                                                                     |
|       LOP       : 19TCLC_NHAT1                                                                                          |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                          |
|                                                                                                                         |
+-------------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
Chuoi S : AGCAATTCAT
Chuoi T : ACAT
n=10,m=4,K =1
So lan xuat hien chuoi T trong chuoi S la :3
```

Hình 12: Giao diện chính của chương trình khi nhập từ file đúng

- Khi dữ liệu đầu vào không hợp lệ . Thì chương trình sẽ thông báo đầu vào không hợp lệ và không giải quyết bài toán nữa .

```
+-------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                             |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)            |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM : NGUYEN MINH QUANG                                                                         |
|                         HO BA THANH                                                                               |
|       LOP       : 19TCLC_NHAT1                                                                                    |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                    |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Chuoi T khong hop le
```

Hình 13: Giao diện chính của chương trình khi lỗi nhận là chuỗi

```
+-------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                             |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)            |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM : NGUYEN MINH QUANG                                                                         |
|                         HO BA THANH                                                                               |
|       LOP       : 19TCLC_NHAT1                                                                                    |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                    |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Tham so khong hop le
```

Hình 14: Giao diện chính của chương trình khi tham số đầu vào không hợp lệ

4.3.2 Kết quả thực thi của chương trình

- Test 1:Với file input có dữ liệu đầu vào là chuỗi S có độ dài là 10, chuỗi T có độ dài là 4 ,với 2 chuỗi cách nhau nhiều nhất là k=1.

```
+-------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                             |
|                          TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)            |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                                     |
|                             HO BA THANH                                                                           |
|       LOP                 : 19TCLC_NHAT1                                                                          |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                        |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
Chuoi S : AGCAATTCAT
Chuoi T : ACAT
n=10,m=4,K =1
So lan xuat hien chuoi T trong chuoi S la :3
```

Hình 15: Kết quả chương trình khi sử dụng thuật toán FFT với Test 1

```
+-------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                               |
|                  TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)              |
|                                     Thuat Toan Vet Can (Brute Force)                                        |
+-------------------------------------------------------------------------------------------- ----------------+
|                                                                                                             |
|       THANH VIEN NHOM         : HO BA THANH                                                                 |
|                                 NGUYEN MINH QUANG                                                           |
|       LOP                     : 19TCLC_NHAT1                                                                |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                              |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
-----------------------
n= 10,m= 4,k= 1
Chuoi S= AGCAATTCAT
Chuoi T= ACAT
So lan xuat hien cua chuoi T trong chuoi S la: 3
```

Hình 16: Kết quả chương trình khi sử dụng thuật toán Vét Cạn với Test 1

➔ Chương trình sử dụng hai thuật toán FFT và Vét Cạn đều cho ra cùng 1 kết quả đúng với file test mẫu đề cho.

- Test 2:Với file input có dữ liệu đầu vào là chuỗi S có độ dài là 100 , chuỗi T có độ dài là 15 ,với 2 chuỗi cách nhau nhiều nhất là k=9.

```
+-------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                               |
|                     TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)           |
+-------------------------------------------------------------------------------------------------------------+
|                                                                                                             |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                               |
|                             HO BA THANH                                                                     |
|       LOP                 : 19TCLC_NHAT1                                                                    |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                  |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
Chuoi S : GTCAGACCCAGGGTGTTCTATACGACGACTTCTCGATTTCTTCGGAGTGATACGGTTGGCATCCACACAACGCGCCCGGAGTACAACATAATCTGCCCGG
Chuoi T : GTCAGACCCAGGGTG
n=100,m=15,K =9
So lan xuat hien chuoi T trong chuoi S la :84
```

Hình 17: Kết quả chương trình khi sử dụng thuật toán FFT với Test 2

```
+-------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                               |
|                  TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)              |
|                                     Thuat Toan Vet Can (Brute Force)                                        |
+-------------------------------------------------------------------------------------------- ----------------+
|                                                                                                             |
|       THANH VIEN NHOM         : HO BA THANH                                                                 |
|                                 NGUYEN MINH QUANG                                                           |
|       LOP                     : 19TCLC_NHAT1                                                                |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                              |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
-----------------------
n= 100,m= 15,k= 9
Chuoi S= GTCAGACCCAGGGTGTTCTATACGACGACTTCTCGATTTCTTCGGAGTGATACGGTTGGCATCCACACAACGCGCCCGGAGTACAACATAATCTGCCCGG
Chuoi T= GTCAGACCCAGGGTG
So lan xuat hien cua chuoi T trong chuoi S la: 84
```

Hình 18: Kết quả chương trình khi sử dụng thuật toán Vét Cạn với Test 2

➔ Cả hai thuật toán tìm chuỗi T trong chuỗi S sử dụng FFT và vét cạn đều cho ra cùng một kết quả.

- Test 3:Với file input có dữ liệu đầu vào là chuỗi S có độ dài là 1000 , chuỗi T có độ dài là 150 ,với 2 chuỗi cách nhau nhiều nhất là k=30.

```
+-------------------------------------------------------------------------------------------------------------------+
|                                                DO LAP TRINH TINH TOAN                                             |
|                     TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                 |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                                     |
|                             HO BA THANH                                                                           |
|       LOP                 : 19TCLC_NHAT1                                                                          |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                        |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
n=1000,m=150,K =30
So lan xuat hien chuoi T trong chuoi S la :851
```

Hình 19: Kết quả chương trình khi sử dụng thuật toán FFT với Test 3

```
+-------------------------------------------------------------------------------------------------------------------+
|                                        DO LAP TRINH TINH TOAN                                                     |
|                  TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                    |
|                                    Thuat Toan Vet Can (Brute Force)                                               |
+-------------------------------------------------------------------------------------------- ----------------------+
|                                                                                                                   |
|       THANH VIEN NHOM          : HO BA THANH                                                                      |
|                                  NGUYEN MINH QUANG                                                                |
|       LOP                      : 19TCLC_NHAT1                                                                     |
|       GIAO VIEN HUONG DAN      : PHAM MINH TUAN                                                                   |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
-----------------------
n= 1000,m= 150,k= 30
So lan xuat hien cua chuoi T trong chuoi S la: 851
```

Hình 20: Kết quả chương trình khi sử dụng thuật toán Vét Cạn với Test 3

➔ Cả hai thuật toán tìm chuỗi T trong chuỗi S sử dụng FFT và vét cạn vẫn cho ra cùng một kết quả.

- Test 4:Với file input có dữ liệu đầu vào là chuỗi S có độ dài là 200000 , chuỗi T có độ dài là 150 ,với 2 chuỗi cách nhau nhiều nhất là k=8000.

```
+-------------------------------------------------------------------------------------------------------------------+
|                                          DO LAP TRINH TINH TOAN                                                   |
|                    TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                                     |
|                             HO BA THANH                                                                           |
|       LOP                 : 19TCLC_NHAT1                                                                          |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                        |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
n=200000,m=150,K =800
So lan xuat hien chuoi T trong chuoi S la :199851
```

Hình 21: Kết quả chương trình khi sử dụng thuật toán FFT với Test 4

```
+-------------------------------------------------------------------------------------------------------------------+
|                                          DO LAP TRINH TINH TOAN                                                   |
|                    TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
|                                        Thuat Toan Vet Can (Brute Force)                                           |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM         : HO BA THANH                                                                       |
|                                 NGUYEN MINH QUANG                                                                 |
|       LOP                     : 19TCLC_NHAT1                                                                      |
|       GIAO VIEN HUONG DAN     : PHAM MINH TUAN                                                                    |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
-----------------------
n= 200000,m= 150,k= 800
So lan xuat hien cua chuoi T trong chuoi S la: 199851
```

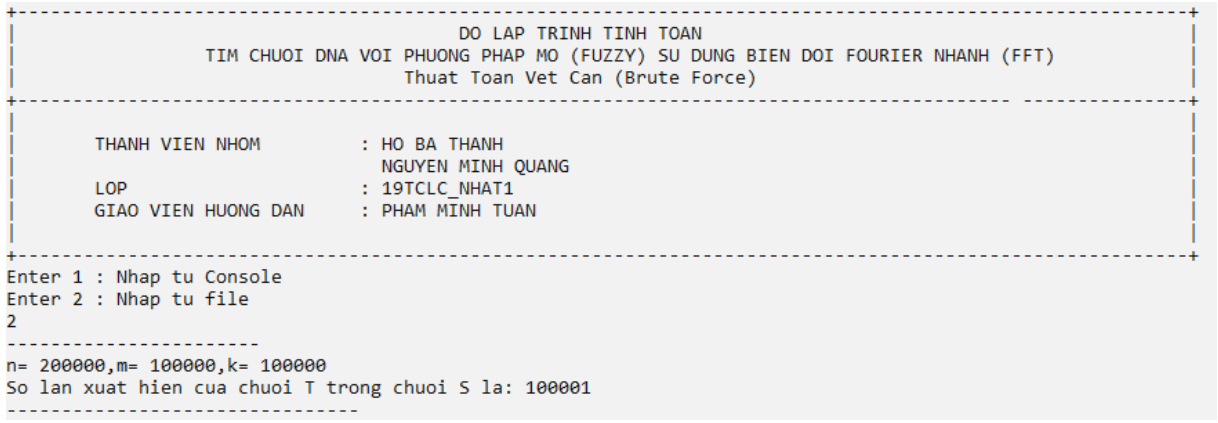
Hình 22: Kết quả chương trình khi sử dụng thuật toán Vét Cạn với Test 4

➔ Cả hai thuật toán tìm chuỗi T trong chuỗi S sử dụng FFT và vét cạn vẫn cho ra cùng một kết quả.Nhưng thời gian chạy của chương trình vét cạn lâu hơn thời gian chạy của chương trình sử dụng FFT.

- Test 5 :Với file input có dữ liệu đầu vào là chuỗi S có độ dài là 200000 , chuỗi T có độ dài là 10000 ,với 2 chuỗi cách nhau nhiều nhất là k=10000.

```
+-------------------------------------------------------------------------------------------------------------------+
|                                          DO LAP TRINH TINH TOAN                                                   |
|                    TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER NHANH (FFT)                  |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
|       THANH VIEN NHOM     : NGUYEN MINH QUANG                                                                     |
|                             HO BA THANH                                                                           |
|       LOP                 : 19TCLC_NHAT1                                                                          |
|       GIAO VIEN HUONG DAN : PHAM MINH TUAN                                                                        |
|                                                                                                                   |
+-------------------------------------------------------------------------------------------------------------------+
Enter 1 : Nhap tu Console
Enter 2 : Nhap tu file
2
Dau vao hop le
n=200000,m=100000,K =100000
So lan xuat hien chuoi T trong chuoi S la :100001
```

Hình 23: Kết quả chương trình khi sử dụng thuật toán FFT với Test 5



Hình 24: Kết quả chương trình khi sử dụng thuật toán Vét Cạn với Test 5

➔ Cả hai thuật toán tìm chuỗi T trong chuỗi S sử dụng FFT và vét cạn vẫn cho ra cùng một kết quả.Nhưng thời gian chạy của chương trình vét cạn lâu hơn rất nhiều thời gian chạy của chương trình sử dụng FFT.So với các Test trước

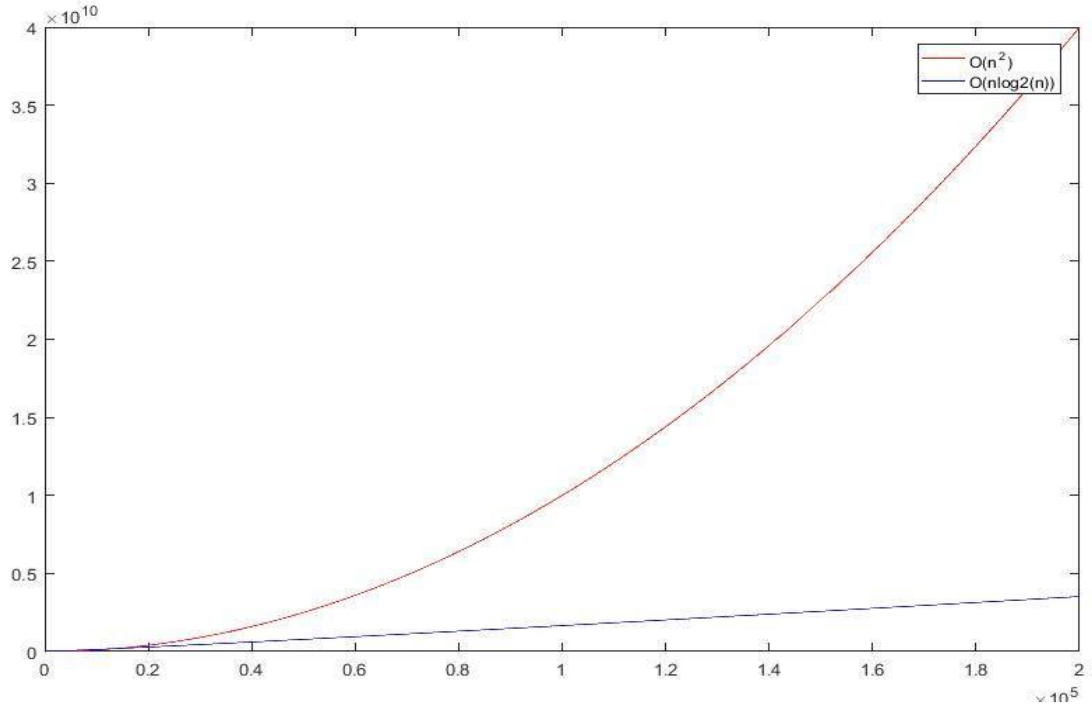
4.3.3 Nhận xét đánh giá

Bảng 2: Bảng phân tích độ tối ưu của hai thuật toán

(Vì mỗi lần chọn Test của chuỗi S,T là bất kỳ → thời gian chạy trong từng trường hợp chọn có thể khác nhau)

Test	**Thông Số**	**Vét Cạn**	**FFT**
		Độ Phức Tạp O(nmk)	**Độ Phức Tạp O((n+m)log(n+m))**
1	**n=10, m=4 k=1**	O(40)	O(16)
2	**n=100, m=15 k=9**	O(13500)	O(237)
3	**n=1000, m=150 k=30**	$O(4{,}5.10^6)$	O(3520)

4	**n=2. 10^5,m=150 k=800**	O(2,4. 10^{10})	O(1,06. 10^6)
5	**n=2. 10^5 m=10^5,k=10^5**	O(2.10^{15})	O(1,64. 10^6)



Hình 25: Đồ thị thể hiện độ phức tạp 2 thuật toán khi n≈m

- Kết quả của chương trình đúng với tất cả bộ test (các bộ test này được kiểm tra bằng phương pháp vét cạn)
- Độ phức tạp thuật toán tìm kiếm chuỗi T trong S bằng FFT là : O((n+m)log(n+m)) trong đó n,m là độ dài của S và T
- Độ phức tạp thuật toán tìm kiếm chuỗi T trong S bằng vét cạn là : O(nmk)
- Với những chuỗi S,T có độ dài ngắn thì cả chương trình sử dụng vét cạn và FFT đều cho ra cùng kết quả .

➔ Tuy nhiên khi chuỗi S và chuỗi T,k rất lớn (khoảng 10^6) lúc đó chương trình vét cạn có độ phức tạp quá lớn . Bởi vì trong trường hợp xấu nhất này độ phức tạp thuật toán vét cạn lúc này tương đương $O(10^{15})$. Trong khi đó với chương trình FFT thì độ phức tạp này chỉ khoảng $O(2.10^6)$ – có thể chạy được trong khoảng 1s .

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau thời gian thực hiện đồ án chúng em đã đạt được nhiều tiến bộ cả về mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình.

- Hiểu biết nhiều hơn về các kỹ thuật và kỹ năng trong lập trình c++.
- Tự tin,phát triển khả năng làm việc nhóm .
- Hiểu hơn 1 phần đề tài nghiên cứa về DNA ứng dụng trong tìm chuỗi xấu hiện nay.
- Cùng với sự hạn chế của thuật toán tìm kiếmVét Cạn,Nhóm đã đưa ra được thuật toán tìm kiếm chuỗi với phương pháp mờ Fuzzy dùng biến đổi FFT để giải quyết bài toán với độ phức tạp lớn,cũng như đột biến lớn.

5.2 Hướng phát triển

- Tiếp thu ý kiến để đồ án được hoàn thiện.

- Tiếp tục phát triển thuật toán,để tối ưu thời gian chạy của từng trường hợp cụ thể.Áp dụng các trường hợp thực tiễn tính toán chuỗi mà lâu nay các phương pháp bình thường không làm được.

TÀI LIỆU THAM KHẢO

[1] Link: https://vi.wikipedia.org/wiki/DNA

[2] Link: https://vi.wikipedia.org/wiki/Logic_mờ

[3] Link: https://vnoi.info/wiki/translate/topcoder/Planning-an-Approach-to-a-Topcoder-Problem-Part-2

[4] Đại học bách khoa thành phố Hồ Chí Minh , bài giảng xử lí tín hiệu số

[5] Chuyên đề tổ hợp – Diễn đàn Mathscope http://mathscope.org/showthread.php?t=43799

[6] Cp-Algorithms, Link: https://cp-algorithms.com/algebra/fft.html

[7] VNOI Wiki, Link: http://vnoi.info/wiki/algo/trick/FFT.md

PHỤ LỤC

```
// Chương trình dùng thuật toán FFT
#include<stdio.h>
#include<iostream>
#include<math.h>
using namespace std;
const int N=600000;       // Khởi tạo độ dài mảng kí tự
const double pi=acos(-1); // Số PI

struct complex // Khởi tạo cấu trúc số phức
{
    double r,i;
    complex(double R=0.0,double I=0.0) { r=R,i=I;}
};

complex operator +(complex a,complex b) { //Toán tử cộng 2 số phức
    return complex(a.r+b.r,a.i+b.i);}
complex operator -(complex a,complex b) { //Toán tử trừ  2 số phức
    return complex(a.r-b.r,a.i-b.i);}
complex operator *(complex a,complex b) { //Toán tử nhân 2 số phức
    return complex(a.r*b.r-a.i*b.i,a.r*b.i+a.i*b.r);}
complex a[N],b[N]; //Khởi tạo mảng số phức
char s[N],t[N];        //Khởi tạo mảng kí tự
int res[N],n,m,k;      //Khởi tạo biến lưu số kí có thể khớp,độ dài chuỗi S,T,k
bool isValid;          //Biến để bắt lỗi tham số đầu vào

// Thuật toán đảo Bit được sử dụng FFT
void reverseBIT(complex *a,int n) //  Đầu vào mảng số phức,độ dài dãy
{
       for (int i=1,j=n/2;i<n-1;i++)
       {
       if (i<j) swap(a[i],a[j]);  // Nếu i là đảo ngược bit của j
       int p=n/2;
       while (j>=p) j-=p,p/=2;  // Sinh ra số nhị phân tiếp theo
       j+=p;
       }
}
```

```
// Thuật toán FFT
void fft(complex *a,int n,int invert) //  Đầu vào mảng số phức,độ dài dãy,biến kiểm FFT hay IFFT
{
        reverseBIT(a,n);   // Đảo ngược Bit để thực hiện biến đổi FFT
        for (int j=2;j<=n;j*=2)
            {
            complex wn(cos(2*pi/j* invert),sin(2*pi/j* invert));
            for (int i=0;i<n;i+=j)
            {
                 complex w(1,0);
                 for (int k=i;k<i+j/2;k++)
                   {
                   complex u=a[k],t=a[k+j/2]*w;
                   a[k]=u+t;a[k+j/2]=u-t;
                   w=w*wn;
                    }
             }
         }
        if (invert ==-1)    //Nếu -1 là biến đổi IFFT
        for (int i=0;i<n;i++) a[i].r/=n;
}

//Thuật toán chuyển mỗi kí tự trong chuỗi cần phân tích và chuỗi cần tìm sang dãy 0,1
void setStringtobinary(char ch) // Đầu vào kí tự (A,C,G,T)
{
      int cnt=0 ;          // Biến lưu số kí tự có thể khớp tại mỗi vị trí
      for (int i=1;i<=k;i++) cnt+=s[i]==ch; // Lưu các vị trí có thể khớp từ kí tự thứ nhất đến kí tự thứ k
      for (int i=1;i<=n;i++) // Với mỗi kí tự tại vị trí i
      {
          if (i+k<=n) cnt+=s[i+k]==ch; // Nếu cách nó một khoảng k có kí tự đang xét,cộng thêm 1
          if (cnt) a[i].r =1;  //Xét vị trí hiện tại,nếu có kí tự đang xét đang cách nó không quá k thì đặt vị trí đó đặt
    bằng 1
          if (i>k) cnt -=s[i-k]==ch; //Xét vị trí hiện tại ,nếu có kí tự đang xét cách nó một khoảng k thì ta bớt đi 1
       }
       for (int i=1;i<=m;i++) b[i].r =t[m+1-i]==ch;  //Chuyển dãy cần tìm sang dãy 0,1
}
```

```
// Thuật toán nhân đa thức nhanh cho mỗi kí tự
void multiplePoly(char ch) // Đầu vào kí tự (A,C,G,T)
{
        int nn=1;
        while (nn<=n+m) nn*=2; // Khởi tạo bậc đa thức cần tính
        for (int i=0;i<nn;i++) a[i].r=a[i].i=b[i].r=b[i].i=0; // Với kí tự đang xét,ta set lại dãy a,b
        setStringtobinary(ch); // Với kí tự đang xét,ta set chuỗi S và T thành dãy 0,1
        fft(a,nn,1);fft(b,nn,1); // Biến đổi FFT 2 dãy a,b với độ dài nn
        for (int i=0;i<nn;i++) a[i]=a[i]*b[i]; // Phép nhân trên miền tần số
        fft(a,nn,-1); // Biến đổi ngược IFFT và thu được hệ số đa thức cần tìm
        for (int i=0;i<nn;i++) res[i]+=(int)(a[i].r+0.1); // Với kí tự đang xét, với mỗi vị trí ta cộng thêm vào biến
        a[i].r để lưu số kí tự có thể khớp tại vị trí đó
}
// Hàm bắt lỗi tham số đầu vào
void catch_error()
{
      int nn=n,mm=m ;
      if(n<0 || m<0 || k<0) {
            cout << "Tham so khong hop le \n";
            return ;
      }
       if(s[n+1] != '\0'){
            cout << "Chuoi S khong hop le \n" ;
            return;}
       if(t[m+1] != '\0'){
            cout << "Chuoi T khong hop le \n" ;
            return;}
       for(int i=1;i<=n;i++) {
            if(s[i]=='A' || s[i]== 'G' || s[i]== 'C' || s[i]== 'T') continue ;
            else nn-=1 ;
       }
       for(int i=1;i<=m;i++) {
           if(t[i]=='A' || t[i]== 'G' || t[i]== 'C' || t[i]== 'T' ) continue;
           else mm-=1 ;
       }
```

```
        if(nn<n) {cout << "Chuoi S khong hop le \n" ; return;}
        if(mm<m) {cout << "Chuoi T khong hop le \n" ; return; }
        if(nn==n && mm==m) {cout << "Dau vao hop le\n"; isValid=true;}
}
// Hàm nhập dữ liệu
void input()
{
        int choice ;
        while(true){
       cout << "Enter 1 : Nhap tu Console \n" ;
        cout << "Enter 2 : Nhap tu file \n";
        scanf("%d",&choice);
        if(choice == 1 || choice == 2) break ;
                else cout << "Lua chon khong hop le\n";
        }
         if(choice==2){
                freopen("input.txt", "r", stdin);
                scanf("%d%d%d",&n,&m,&k);
                scanf("%s",s+1);scanf("%s",t+1);
                catch_error();
          }
          else{
                printf("n="); scanf("%d",&n);
                printf("m="); scanf("%d",&m);
                printf("k="); scanf("%d",&k);
                printf("Chuoi S :"); scanf("%s",s+1);
                printf("Chuoi T :"); scanf("%s",t+1);
                catch_error();
           }
}
```

```
// Hàm xử lý kết quả
void solve()
{
	if(isValid==true){	// Nếu dữ liệu nhập vào đúng
	//Lưu số ký tự có thể khớp tại mỗi vị trí
	multiplePoly('A');multiplePoly('C');
	multiplePoly('G');multiplePoly('T');
	long long ans=0;  //Kết quả
	for (int i=m+1;i<=n+1;i++)
	if (res[i]==m) ans+=1; //Nếu tại ví trị i có đúng m ký tự khớp thì +1
	printf("Chuoi S : %s\n",s+1);
	printf("Chuoi T : %s\n",t+1);
	printf("n=%d,m=%d,K =%d\n",n,m,k);
	cout << "So lan xuat hien chuoi T trong chuoi S la :";
	cout<<ans<<endl;
	}
}
int main()
{
	cout << "+------------------------------------------------------------+" << endl;
	cout << "|                        DO LAP TRINH TINH TOAN              |" << endl;
	cout << "| TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI FOURIER
	NHANH (FFT) |" << endl;
	cout << "+------------------------------------------------------------+" << endl;
	cout << "|                                                            |" << endl;
	cout << "|      THANH VIEN NHOM      : NGUYEN MINH QUANG              |" << endl;
	cout << "|                             HO BA THANH                    |" << endl;
	cout << "|      LOP                  : 19TCLC_NHAT1                   |" << endl;
	cout << "|      GIAO VIEN HUONG DAN : PHAM MINH TUAN                  |" << endl;
	cout << "|                                                            |" << endl;
	cout << "+------------------------------------------------------------+" << endl;
	input();
	solve();
	}
```

```
// Chương trình sử dụng thuật toán Vét Cạn(Brtute Force)
#include<stdio.h>
#include <iostream>
using namespace std ;
int n,m,k ,ans,cnt;
string s,t ;

// Hàm Vét Cạn
void BruteForce()
{
        s=" "+s ;
        t=" "+t ;
        int cnt=0;
        for(int i=1 ;i<=n-m+1;i++) //Chạy so khớp với chuỗi S với(n-m+1)lần
        {
            int ans=0; //Chạy biến đếm ban đầu là 0
            for(int j=1;j<=m;j++) //Lấy từng ký tự của chuỗi T so sánh
           {
                char temp=t[j];
                for(int u=max(i+j-1-k,0);u<=min(i+j-1+k,n);u++)
                //So sánh ký tự của chuỗi T trong khoảng cho phép k
               {
                if(temp==s[u] && u >= 1 && u <= n)
                {
                        ans+=1 ; //Nếu đúng thì ta tăng biến đếm
                        break ;
                }
            }
         }
         if(ans==m) cnt+=1 ;
         //Nếu so sánh tất cả các ký tự đều đủ đk thì tăng sô lần suất hiện
        }
        cout<<cnt;
}
```

```
// Hàm menu của chương trình
void menu()
{       int choice;
        while(true){
        cout << "Enter 1 : Nhap tu Console \n" ;
        cout << "Enter 2 : Nhap tu file \n";
        cin>>choice;
        if (choice != 1 && choice != 2){
                cout << "\nNhap sai!!! Moi Chon Lai: ";
                cin >> choice ;
        }
        cout << "- - - - - - - - - - - - - - - - - - - - - - - - \n";
        fflush(stdin);
        switch (choice)
        {
                case 1 : { //Nhập từ console
                 cout<<"n= ";cin>>n;
                 cout<<"m= ";cin>>m;
                 cout<<"k= ";cin>>k;
                 cout<<"Chuoi S= ";cin>>s;
                 cout<<"Chuoi T= ";cin>>t;
                 cout<<"So lan xuat hien cua chuoi T trong chuoi S la: " ;
                 BruteForce();
                 break;
                 }
                 case 2 : { //Đọc từ file dữ liệu
                 freopen("input.txt", "r", stdin);
                 cin>>n>>m>>k>>s>>t;
                 cout<<"n= "<<n<<","<<"m= "<<m<<","<<"k= "<<k<<endl;
                 cout<<"Chuoi S= "<<s<<endl;
                 cout<<"Chuoi T= "<<t<<endl;
                 cout<<"So lan xuat hien cua chuoi T trong chuoi S la: ";
                 BruteForce();
                 break;
                 }
        }break;
        }
}
```

```
int main()
{
   cout << "+- - - - - - - - - - - - - - - - - - - -  - - - -- - - - - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - +" << endl;
   cout << "|                                   DO LAP TRINH TINH TOAN                                  |" << endl;
   cout << "| TIM CHUOI DNA VOI PHUONG PHAP MO (FUZZY) SU DUNG BIEN DOI
FOURIER NHANH (FFT)                                                                                   |" << endl;
   cout << "|                                 Thuat Toan Vet Can (Brute Force)                           |" << endl;
   cout << "+- - - - - - - - - - - - - - - - - - - -  - - - -- - - - - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - +" << endl;
   cout << "|                                                                                            |" << endl;
   cout << "|              THANH VIEN NHOM         : HO BA THANH                                         |" << endl;
   cout << "|                                        NGUYEN MINH QUANG                                   |" << endl;
   cout << "|              LOP                     : 19TCLC_NHAT1                                        |" << endl;
   cout << "|             GIAO VIEN HUONG DAN  : PHAM MINH TUAN                                          |" << endl;
   cout << "|                                                                                            |" << endl;
   cout << "+- - - - - - - - - - - - - - - - - - - -  - - - -- - - - - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - +" << endl;
   menu();
}
```