תרגיל מסכם ("פרויקט") בתכנות מקבילי ומבוזר סמסטר קיץ 2021

התרגיל יכתב בעזרת MPI, OpenMP, CUDA.

תאור הבעיה של sequence alignment

התרגיל עוסק בהשוואה בין סדרות של אותיות באי"ב האנגלי.

הוא פישוט של אלגוריתמים שמשתמשים בהם בביו-אינפורמטיקה שם המטרה היא למצוא דמיון בין מולקולות ביולוגיות (חלבונים, DNA, RNA). בהקשר זה כל אות מייצגת איזו ישות כימית וסדרה של אותיות מייצגת מבנה של מולקולה.

למשל מולקולת DNA מורכבת מארבעה סוגים של nucleotides שמקובל לסמן אותם באותיות A, T, C, G. ואז המחרוזת TCCGT יכולה לייצג קטע של מולקולת DNA.

הבעיה של השוואת סדרות שנעסוק בה מכונה בביולוגיה בעיית ה- **sequence alignment**.

הגדרה של alignment score.

כשמשווים בין 2 סדרות, נותנים "ציון" לתוצאת ההשוואה. זה ה- alignment score.

ככל שהציון גבוה יותר -- הסדרות נחשבות לדומות יותר.

חישוב ה- alignment score של שתי סדרות שאורכן שווה

במהלך ההשוואה נשווה בין כל אות בסדרה הראשונה לאות המתאימה בסדרה השניה.

ליד כל זוג אותיות נסמן את תוצאת ההשוואה בעזרת סימן בודד:

-- אם האותיות זהות נסמן את הזוג עם סימן הדולר ($)

-- אם האותיות אינן זהות אבל שתיהן שייכות לאותה קבוצה מבין הקבוצות הבאות (אותן נכנה "קבוצות מסוג ראשון") אז נסמן את הזוג בסימן האחוז "%".

```
NDEQ        NEQK        STA
MILV        QHRK        NHQK
FYW         HY          MILF
```

למשל זוג האותיות (S, A) יסומן בסימן % כי שתיהן שייכות לקבוצה STA.

זוג האותיות (E, N) יסומן גם הוא בסימן % כי שתי האותיות שייכות לקבוצה NDEQ. (שתיהן גם שייכות לקבוצה NEQK וזו סיבה נוספת לסמן אותן ב- %).

-- אם האותיות אינן זהות והן אינן שייכות לאותה קבוצה מסוג ראשון אבל הן כן שייכות לאותה קבוצה מבין הקבוצות הבאות ("קבוצות מסוג שני") אז נסמן את הזוג בסימן הסולמית (#).

```
SAG      ATV      CSA
SGND     STPA     STNK
NEQHRK   NDEQHK   SNDEQK
HFY      FVLIM
```

לדוגמא זוג האותיות AV יסומנו בסימן # כי שתיהן נמצאות בקבוצה ATV (ואין קבוצה מסוג ראשון אליה שייכים שתי האותיות האלו).

-- אם האותיות אינן זהות ואין קבוצה מסוג ראשון או שני אליה שייכות שתיהן אז הזוג יסומם בסימן הרווח (' ').

אחרי סימון כל הזוגות, משתמשים בסדרת הסימנים שנוצרה כדי לחשב את ה- alignment score של שתי המחרוזות לפי הנוסחה הבאה:

alignment score =

$$W_1 * \text{numberOfDollars} - W_2 * \text{NumberOfPercents} - W_3 * \text{NumberOfHashes} - W_4 * \text{NumberOfSpaces}$$

כאן W1, W2, W3, W4 הם המשקלים שיש לתת לכל אחד מהרכיבים של הנוסחה. אלו הם מספרים שלמים לא שליליים שיופיעו בקלט (ראו בהמשך).

<u>דוגמא</u>

נניח w1=10, w2=2, w3=3, w4=4

נחשב את ציון ההשוואה של שתי סדרות של אותיות:

Seq1 =	A	P	Q	R	S	B	A	T	A	V
Seq2 =	A	S	Q	R	S	E	A	V	S	L
Symbols=	$	#	$	$	$		$	#	%	%

alignment score = 10 * 5 – 2 * 2 – 3 * 2 – 4 * 1 = 36

<u>חישוב ה- alignment score של 2 מחרוזות Seq1, Seq2 כאשר Seq2 קצרה יותר.</u>

רושמים את Seq2 מתחת ל- Seq1 בהיסט (offset) מסוים

כאשר offset >= 0. אסור שאותיות של Seq2 יופיעו מעבר לסוף של Seq1.

את ה- alignment score מחשבים לפי האלגוריתם הנ"ל כאשר מתעלמים מהאותיות של הסדרה הארוכה יותר שאין להן אות תואמת.

<u>דוגמא</u>

נניח: w1=10, w2=2, w3=3, w4=4

Seq1 =	A	P	Q	R	S	B	A	A	V	V
Seq2 =				R	S	T	B	T	L	
Symbols=	–	–	–	$	$			%	%	–

alignment score = 10* 2 – 2* 2 – 3*0 –4*2 = 8

כאן offset=3. תווים שלא נלקחו בחשבון סומנו כאן במקף (–).

אם נשנה את ה- offset לאחד החישוב יהיה:

Seq1 =	A	P	Q	R	S	B	A	A	V	V
Seq2 =		R	S	T	B	T	L			
Symbols=	–		#					–	–	–

alignment score = 10*0 – 2*0 - 3*1 –4*5 = –23

הגדרה של Mutant Sequence ("מוטציה")

עבור סדרת אותיות seq נגדיר את ה- Mutant Sequence שיסומן ב- MS(k) כסדרת האותיות המתקבלת ע"י הוספת מקף (hyphen) אחרי הסימן ה- k – י ב- seq כאשר k = 1, 2, … (strlen(seq)). נגדיר גם שהסדרה המקורית seq היא "מוטציה" של עצמה.

לדוגמא עבור סדרת האותיות ABCAT יהיו חמישה mutant sequences:

MS(1) = A–BCAT

MS(2) = AB–CAT

MS(3) = ABC–AT

MS(4) = ABCA–T

MS(5) = ABCAT–

"המוטציה" האחרונה היא בעצם הסדרה המקורית ABCAT עצמה.

כאשר משווים mutant sequence לסדרה אחרת אז המקף (והאות התואמת לו בסדרה האחרת) אינם נלקחים בחשבון בחישוב של ה- alignment score.

(במקרה שהמקף מופיע בסוף המוטציה כמו ב- MS(5) בדוגמא כאן אז ניתן להשמיט את המקף כאשר משווים את המוטציה לסדרה אחרת).

תאור הבעיה

עבור מחרוזות נתונות Seq1, Seq2 כאשר Seq2 היא הקצרה יותר יש למצוא את ה- offset (שנסמנו n) ואת המוטציה MS(k) של Seq2 עבורם יתקבל ה- alignment score המקסימלי כשמשווים את המוטציה של Seq2 ל- Seq1.

דוגמא

w1=10, w2=2, w3=3, w4=4

seq1 = HELLOWORLD

seq2 = OWRL

אז ה- alignment score המקסימלי יתקבל עבור n = 4

ו- k =2:

HELLOWORLD

OW–RL

<u>התכנית</u>

בקלט יופיעו מספר סדרות של אותיות. יש להשוות את הסדרה הראשונה (נסמנה Seq1) לכל אחת מהסדרות המופיעות בהמשך הקלט. עבור כל אחת מסדרות אלו יש למצוא את ה- offset (n) ואת מיקום המקף במוטציה (k) עבורן יתקבל ה- alignment score המקסימלי כשמשווים את המוטציה ל- Seq1.

הקלט לתוכנית יופיע ב- standard input. הפלט יכתב ל- standard output.

רק אחד מתהליכי ה- MPI יקרא את הקלט וידאג להעביר את המידע הנחוץ לתהליכים האחרים. אותו תהליך יכתוב גם את הפלט.

מספר תהליכי ה- MPI יקבע בעת הרצת התכנית. אין להניח שיהיו רק 2 תהליכים.

גם מספר ה- threads של OpenMP לא צריך להיות קבוע בקוד של התכנית

(אין להגדיר משהו כמו #define NTHREADS 4).

<u>טיפ</u>: כשמשווים בין 2 אותיות במהלך חישוב ה- alignment score, רצוי להיעזר במבנה נתונים כדי להאיץ את מציאת המשקל שיש לתת לתוצאת ההשוואה. כאן מבנה נתונים פשוט יהיה אפקטיבי (לא צריך hash table). חיפוש סדרתי פשוט של 2 האותיות המושוות בקבוצות "מסוג ראשון" ובקבוצות "מסוג שני" יגרום לתכנית לרוץ לאט מאוד.

<u>הפורמט של הקלט</u>

בשורה הראשונה יופיעו 4 המשקלים w1, w2, w3, w4.

בשורה הבאה תופיע הסדרה Seq1 (לא יותר מ- 3000 אותיות)

בשורה הבאה יופיע מספר שלם שנכנה אותו כאן number_of_sequences.

זה מספר הסדרות שיש להשוות לסדרה הראשונה (Seq1).

ב- number_of_sequences השורות הבאות יופיעו הסדרות אותן יש להשוות

ל- Seq1 כאשר כל סדרה כזאת תופיע בשורה נפרדת. האורך של כל סדרה כזאת לא יעלה על 2000 אותיות ואורכה יהיה קטן מהאורך של Seq1.

<u>הפלט</u>

בפלט יופיעו number_of_sequences שורות, שורה עבור כל סדרה Seq2 שהופיעה בקלט והושוותה לסדרה הראשונה Seq1. בשורה ירשם מה ה- offset (n)

ומיקום המקף (k) במוטציה של Seq2 שמניבה את ה- score המקסימלי בהשוואה ל- Seq1. כל שורה תהיה בפורמט k = ... n = ... הסדר של השורות בפלט תואם את סדר הופעת הסדרות Seq2 בקלט.

<u>הנחיות</u>

התרגיל יכתב תוך שימוש ב- MPI, OpenMP ו- Cuda.

ההגשה דרך מודל <u>ביחידים</u>. מותר להתייעץ עם חברים אבל את הקוד יש לכתוב לבד.

אם משתמשים בקוד שהורד מהאינטרנט יש לציין את מקורו.

יש לצרף תיעוד שמסביר את האלגוריתם ואת מבני הנתונים בהם השתמשתם.

יש להסביר גם כיצד מוקבלה התכנית.

כל סטודנט או סטודנטית "יגנו" על העבודה בפגישת zoom בה הם יריצו את התרגיל ויסבירו מה עשו. היו מוכנים להדגים את ריצת התכנית על 2 מחשבים. יש לדעת להסביר כל שורה בקוד.

התוכנית צריכה לרוץ מהר יותר מגרסה סדרתית שלה.

<u>הניקוד</u>:

15 נקודות יורדו אם אין שימוש ב- CUDA.

האלגוריתם, הצורה שבה התכנית מוקבלה ואיכות הקוד ילקחו בחשבון בעת מתן הציון.

בהצלחה!

23 אוקטובר 2021