UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

CI-0117 Programación paralela y concurrente

Grupos 2 y 3

Enunciado tarea programada IV (MPI)

Fecha de entrega: 2025/Nov/28

Modalidad: **grupos de máximo dos personas**

Encontrar similaridades en cadenas de ADN

Debe construir un programa C++ MPI que encuentre la **subsecuencia** común más larga entre dos secuencias de ADN generadas de manera aleatoria y suficientemente grandes, indique su longitud y la despliegue.

Pasos a completa en esta tarea programada:

1. Debe realizar una **investigación** y proponer un algoritmo (pseudo-código) para resolver este problema (30% de la nota de esta tarea), que no sea el de fuerza "bruta", se trata de un documento que debe entregarse al profesor a más tardar el 14/Nov
2. Debe construir una versión serial que resuelva el problema (30% de la nota de esta tarea) que utilice el diseño aprobado del punto anterior
3. También debe construir una versión paralela en **MPI** (40%), por lo que es importante que en su diseño sea posible de paralelizar utilizando esta herramienta

Una **subsecuencia** de un conjunto de caracteres $x_0x_1x_2x_3x_4x_5...x_{n-1}$ es otro conjunto de la forma $x_{j1}x_{j2}x_{j3}$ … x_{jk} donde $j_i < j_{i+1}$. Ejemplo, para la tira "1234567890", las siguientes son **subsecuencias**: "12467", "230", "134579", "24680", etc.

Si tenemos estas dos cadenas de ADN S_1 y S_2 y queremos encontrar la **subsecuencia** común más larga entre ellas:

- S_1 = "ACCGGTCGAGTGCGCGGAAGCCGGCCGAA"
- S_2 = "GTCGTTCGGAATGGCCGTTGCTCTGTAA"
- La solución sería la secuencia S_x = "GTCGTCGGAAGCCGGCCGAA":

```
ACCGGTCGAGTGCGCGGAAGCCGGCCGAA
   | ||  || || | ||||||||||||
   G TC  GT CG G AAGCCGGCCGAA     [Esta es la secuencia común más larga]
   GTCGT CGGAA  GCCG  GC C G AA
   ||||| |||||  ||||  || | | ||
   GTCGTTCGGAATGGCCGTTGCTCTGTAA
```

La solución por fuerza “bruta” sería encontrar todas la subsecuencias de la primera cadena S_1, $O(2^N)$; para cada una de ellas determinar si existe en la segunda cadena S_2, O(N), para luego escoger la más larga de todas las coincidentes. No queremos utilizar esta solución, pues requiere $O(N * 2^N)$ para resolver el problema, donde N representa la longitud de la cadena más larga (complejidad exponencial).

Clases C++ provistas

- **ADN**
  - Representa las cadenas de ADN como tiras de caracteres
  - Posee dos constructores para crear las secuencias de ADN
  - Método para listar todas las subsecuencias de una instancia

Archivos provistos

- Pruebas para la clase ADN: sequences.cc
- Makefile para compilar los códigos provistos

Restricciones adicionales

- Conformar los grupos de trabajo e indicar al profesor los integrantes. Puede haber grupos compuestos por una sola persona
- La solución debe utilizar las estructuras provistas por MPI y correr en los equipos disponibles de la ECCI
- El código presentado debe ser **original**, no puede ser copiado o reformulado de una solución disponible en Internet, el plagio es considerada una falta muy grave en los reglamentos de la UCR
- El **diseño** debe ser presentado al profesor y recibir su **aprobación** para obtener la nota final de esta tarea programada. Revisaremos las propuestas durante las clases del Viernes 14/Nov o antes si lo prefieren
- El programa debe desplegar la subsecuencia encontrada y su longitud
- La solución paralela presentada debe mejorar sustancialmente el rendimiento de la versión serial, por lo que tiene que ser posible generar secuencias de ADN grandes, de varios miles de nucleótidos, igual que en otros ejemplos debe construir una tabla con las mejoras de rendimiento en varios casos

Referencias

- https://es.wikipedia.org/wiki/Problema_de_subsecuencia_común_más_larga