

Algoritmos y Estructuras de Datos II. 2019. Examen de Laboratorio.

El ejercicio consiste en implementar el TAD *DNA* especificado en el examen teórico.

Un DNA es una secuencia no vacía de nucleótidos (*T*, *A*, *C*, *G*). Por ejemplo:

TACCAGG,
A,
CC,

son DNA válidos.

La implementación debe realizarse usando una *lista enlazada de nodos*, donde cada nodo debe contener un único nucleótido.

Las funciones a implementar son las siguientes:

Devuelve un DNA con un único nucleótido de tipo T.
`dna_t dna_T()`

Devuelve un DNA con un único nucleótido de tipo C.
`dna_t dna_C()`

Devuelve un DNA con un único nucleótido de tipo A.
`dna_t dna_A()`

Devuelve un DNA con un único nucleótido de tipo G.
`dna_t dna_G()`

Concatena el segundo DNA al final del primer DNA. Modifica el primer argumento.
`dna_t dna_join(dna_t first, dna_t second)`

Devuelve la longitud del DNA.
`unsigned int dna_length(dna_t dna)`

Imprime en pantalla un DNA.
`void dna_print(dna_t dna)`

Verifica si el primer DNA es prefijo del segundo.
`bool dna_is_prefix(dna_t first, dna_t second)`

Verifica si ambos DNA son iguales (tienen la misma longitud y además uno es prefijo del otro).
`bool dna_is_equal(dna_t first, dna_t second)`

Corta el DNA en dos segmentos disjuntos, el primero de longitud 'count'. Concatenando ambos segmentos se obtiene el DNA original. Se asume que 'count' es mayor estricto que 0 y menor estricto que la longitud del primer DNA. Los dos segmentos se devuelven en un arreglo de tamaño 2.

*`dna_t *dna_cut(dna_t dna, unsigned int count)`*

Destruye el DNA liberando sus recursos de memoria.

dna_t dna_destroy(dna_t dna)

Ejercicios

1. Crear el archivo *dna.c*, implementar allí todas las funciones que se exportan en *dna.h*.
2. Crear el archivo *main.c*, implementar allí una función *main* que ejecute al menos un caso de prueba para todas las funciones anteriores.

No está permitido modificar ningún otro archivo, sólo *dna.c* y *main.c*.

*Se restarán puntos en caso de **memory_leaks** o **invalid_reads** detectados por **valgrind**.*

Compilar: *\$ make*

Ejecutar main: *\$ make run_main*

Ejecutar tests: *\$ make run_tests*

Entregar el final de acuerdo a las instrucciones del profesor a cargo.