

Tarea 2 - Opción 1 – SSS-Index (Sistemas Distribuidos)

Esta tarea consiste en **procesar consultas** usando el índice SSS-Index (implementado en la Tarea 1) mediante programación multi-hilo con OpenMP en lenguaje C. Para esto, se le proporcionará los datos del índice para que los use como entrada de su programa.

Su tarea será revisada utilizando archivos de prueba con el siguiente formato (y usando el operador “<” para redirigir la entrada al archivo de prueba):

test.txt:

```
5
0.137 -0.03 0.09 0.10 0.2 -0.1 -0.05 0.08 0.7 0.6 0.01 0.2 0.4 0.03 0.4 0.6 0.2 0.5 0.6 0.9
-0.3 0.9 0.18 0.4 0.1 0.03 0.77 -0.2 1.3 0.1 0.3 0.1 0.02 0.6 0.7 0.1 -0.8 0.8 0.7 0.083
-0.0 0.2 -0.031 0.24 0.1 0.5 0.7 0.8 -0.1 0.5 -0.0 -0.2 0.1 0.5 0.02 0.06 0.96 0.56 0.16 0.30
0.9 0.5 0.26 0.1 -0.9 0.003 -0.8 0.6 0.03 0.7 0.1 -0.9 0.02 0.3 0.3 0.0 0.7 0.65 0.028 -0.044
0.4 -0.1 0.0 0.1 0.5 -0.0 0.8 0.8 -0.6 0.4 0.27 0.4 0.34 0.5 -0.2 0.82 -0.1 0.3 0.043 0.246
2
0.0 0.9 0.8 0.4 0.1 0.1 0.77 0.2 1.3 0.0
0.137 -0.03 0.09 0.10 0.2 -0.1 -0.05 0.08 0.7 0.6 0.01 0.2 0.4 0.03 0.4 0.6 0.2 0.5 0.6 0.9
0.4 -0.1 0.0 0.1 0.5 -0.0 0.8 0.8 -0.6 0.4 0.27 0.4 0.34 0.5 -0.2 0.82 -0.1 0.3 0.043 0.246
2
0.8 0.7 0.45 0.26 0.1 0.9 0.3 -0.8 0.6 0.3 0.7 0.1 0.9 0.02 0.3 0.3 0.0 -0.7 0.65 0.028 0.4
0.8 0.1 0.3 0.31 0.24 0.1 -0.5 0.7 0.8 0.1 0.5 0.0 -0.2 0.1 0.5 0.02 0.06 0.96 0.5 0.6 0.30
```

En la primera línea, se indica el número de elementos de la base de datos, que en este ejemplo en particular son 5. Desde la segunda línea, se indican los elementos (vectores) de la base de datos, que en este caso son 5 vectores de dimensión 20. Siempre las coordenadas de los vectores estarán separadas por un espacio. Luego, se indica la cantidad de pivotes (2 en este ejemplo en particular). A continuación se indican las distancias de la *Matriz de Distancias*, que en este caso está compuesta por 10 elementos (cantidad de pivotes multiplicado por cantidad de elementos), tal como muestra la Figura 1. **Nótese que las distancias están ordenadas para rellenar una matriz donde las filas representan a los pivotes y las columnas a los elemento de la base de datos** (ver Figura 1). Desde la siguiente línea, se indican los pivotes (uno por línea), que en este caso corresponden al primer y quinto elemento (E_0 y E_4). Luego, se indica el número de consultas a procesar, que deben ser 2 en este caso. Posteriormente están las consultas en sí, las que están formadas por un radio de búsqueda y un vector consulta (también de dimensión 20), por lo tanto, el primer número (0,8) corresponde al radio de la consulta y los demás números de la fila corresponden a las coordenadas del vector consulta. En este ejemplo, ambas consultas tienen un radio de búsqueda igual a 0,8.

	E_0	E_1	E_2	E_3	E_4
P_0	0	0.9	0.8	0.4	0.1
P_1	0.1	0.77	0.2	1.3	0

Figura 1. Matriz de Distancia para el ejemplo de test.txt.

En la Matriz de Distancias de la Figura 1 hay 2 celdas con distancia 0 en este ejemplo, pues para este ejemplo en particular los pivotes corresponden a los elementos E_0 y E_4 .

La dimensión de los vectores (elementos de la base de datos y consultas) siempre será 20. Utilice
Semestre 1/2022

#define DIM 20 para indicar la dimensión de los elementos en su programa en C. Como función de distancia debe usar la distancia **euclidiana**.

El proceso de consultas tiene por objetivo evitar el cálculo de distancia entre la consulta y todos los elementos de la BD. El algoritmo de procesamiento de una consulta (q, r) (donde r es el radio y q el vector de consulta) es el sgte.:

- 1) Calcular y almacenar la distancia entre todos los pivotes y el vector consulta q .
- 2) Intentar descartar cada elemento E_i utilizando las distancias almacenadas entre E_i y todos los pivotes (en la Matriz de Distancias), de la siguiente manera indicada por el Algoritmo 1:

```
for (i=0; i<Size_BD; i++)
{
    flag_descarte = 0;
    for (j=0; j<P; j++) /* P=cantidad de pivotes */
    {
        if (dist(q, Pj)+r < dist(Ei, Pj) || dist(q, Pj)-r > dist(Ei, Pj))
        {
            flag_descarte = 1; /* Ei es descartado */
            break;
        }
    }
    if (flag_descarte == 1)
        continue;

    /*En caso de no poder descartar el elemento con ningún pivote, se realiza el
    cálculo de distancia entre la consulta q y el elemento. */
    if (dist(q, Ei) < r)
        Ei es parte de la respuesta
}
```

Algoritmo 1. Algoritmo de descarte de elementos.

Tenga en cuenta que al ejecutar la sentencia “continue;” en el código, **no** se ejecuta el código entre el continue y el final de la iteración, y se continua con la siguiente iteración inmediatamente.

Su programa debe imprimir por pantalla **solamente 2 cosas**: 1) la **cantidad** de elementos resultado por consulta (no imprima las coordenadas de los vectores), y 2) al final debe imprimir también la **cantidad de elementos descartados en total**, es decir, tomando en cuenta todas las consultas.

Usted deberá distribuir el cómputo entre los distintos hilos de ejecución. Usted decida la estrategia/manera de distribuir el procesamiento de las consultas.

Su tarea debe funcionar con T hilos ($T > 2$). Utilice #define T 8 para definir T como constante en su programa.

Para calificar su tarea, se tomará en cuenta los siguientes ítems:

- 1) Correcta lectura de los datos.
- 2) Correcta distribución del cómputo (descarte y cálculos de distancia).
- 3) Correcta implementación del algoritmo de descarte (Algoritmo 1).
- 4) Impresión por pantalla del resultado correcto.

Notas :

- Plazo de entrega: Por LMS a más tardar el **11 de Julio del 2022**.
- El trabajo es de a 2 personas. Deben ser los mismo equipos que en la Tarea 1. En caso de problema con esta condición, debe dar aviso oportunamente por email.
- Debe abordar la misma opción de Tarea que en la Tarea 1.