10-junio-2025: En el diagrama del B-tree (11-a) se cambia a 61 el valor de una celda que decía 60.

Control 3

CC5213 – Recuperación de Información Multimedia
Departamento de Ciencias de la Computación
Universidad de Chile
Profesor: Juan Manuel Barrios
Fecha entrega: lunes 23 de junio de 2025

Semana 09

1. Tiene un dataset con **10.000** documentos de texto. A continuación, se muestran dos documentos de ese dataset:

   D_1 = Suchai es un satélite artificial

   D_2 = mi perro tiene mucha inteligencia

   Un usuario escribe la siguiente consulta de texto:

   Q = inteligencia artificial

   Se desea identificar cuál de estos dos documentos **D_1** y **D_2** es más similar a la consulta **Q** siguiendo el modelo Bag-of-Words y TF-IDF visto en clases.

   Para esto, realice los siguientes pasos:

a. (0,2 puntos) Calcule el IDF para el vocabulario relevante. La siguiente tabla muestra la frecuencia de documentos para algunas palabras del dataset.

Id	palabra	documentos
1	artificial	1
2	es	1.000
3	inteligencia	10
4	mi	1.000
5	mucha	1.000
6	perro	100
7	satélite	10
8	Suchai	10
9	tiene	1.000
10	un	1.000

b. (0,9 puntos) Calcule el vector TF-IDF normalizado de **D_1**, **D_2** y **Q** siguiendo las fórmulas vistas en clases.

c. (0,2 puntos) Calcule la similitud coseno entre **Q**, **D_1** y **D_2** y señale el documento más similar a **Q**.

d. (0,2 puntos) Calcule la distancia euclidiana entre **Q**, **D_1** y **D_2** y señale el documento más cercano a **Q**.

Algunas fórmulas que pueden ser de utilidad (logaritmos en base 10):

$$idf_i = \log\left(\frac{N}{n_i}\right) \qquad tf_{ij} = \begin{cases} 1 + \log(freq_{ij}) & \text{si } freq_{ij} > 0 \\ 0 & \text{si no} \end{cases}$$

$$||d'_j|| = \sqrt{\sum_{i=1}^{t} w_{ij}^2}$$

(norma de un vector)

$$d_j = \frac{d'_j}{||d'_j||}$$

(normalización a largo 1)

$$cos(q, d_j) = \sum_{i=1}^{t} w_{iq}\, w_{ij}$$

(similitud coseno de vectores normalizados)

Semana 10

1. **N-Grams:** Suponga que tiene un dataset que es tokenizado en **50.000** palabras en total quedando con un vocabulario de **200** palabras distintas.

   a. (0,2 puntos) Cada documento se desea representar por vectores tf-idf utilizando **bigramas**, es decir, los términos a considerar son grupos de n palabras consecutivas, con **n ≤ 2**. Señale la cantidad de términos totales que tendrá el dataset y señale una cota superior de la dimensión de los vectores que serán creados.

   b. (0,2 puntos) Cada documento se desea representar por vectores tf-idf utilizando **trigramas**, es decir, los términos a considerar son grupos de n palabras consecutivas, con **n ≤ 3**. Señale la cantidad de términos totales que tendrá el dataset y señale una cota superior de la dimensión de los vectores que serán creados.

2. **Distancia de edición**: Suponga que en un buscador el usuario ingresa la palabra "salos". El buscador no encuentra esa palabra en el vocabulario, pero tiene dos palabras candidatas que podría utilizar: "zorros" y "salsas". Se usará la distancia de edición o de Levenshtein[1] para determinar cuál de estas dos palabras es la más parecida a la ingresada por el usuario. Esta distancia utilizará una función de costo con las siguientes consideraciones:

   - El costo de inserción de un carácter es 2.
   - El costo de borrado de un carácter es 2.
   - El costo de reemplazo entre z y s es 0.3.
   - El costo de reemplazo entre vocales es 0.5.
   - El costo de reemplazo entre el resto de caracteres es 1.

   a. (0,4 puntos) Calcule la distancia de edición o de Levenshtein entre "salos" y "zorros", incluyendo la matriz de costos asociada.

   b. (0,4 puntos) Calcule la distancia de edición o de Levenshtein entre "salos" y "salsas", incluyendo la matriz de costos asociada.

[1] Ver un ejemplo de cálculo de distancia de Levenshtein en la sección de Material Docente del curso Slides 10.2-Descriptores de texto-Índice invertido.pdf

3. **LSA:** Suponga que tiene un dataset de 20.000 documentos con un vocabulario de de 5.000 términos distintos. Sea A una matriz que contiene en cada fila el descriptor tf-idf de cada documento del dataset.

   a. (0,2 puntos) Señale las dimensiones y explique qué representa la matriz B resultante de la multiplicación $B = A * A^T$ (en máximo dos líneas).

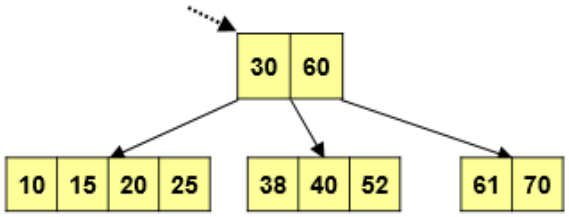
   b. (0,2 puntos) Señale las dimensiones y explique qué representa la matriz C resultante de la multiplicación $C = A^T * A$ (en máximo dos líneas).

   c. (0,2 puntos) Explique la mayor ventaja de LSA con respecto a tf-idf desde el punto de vista de la efectividad (en máximo dos líneas).

   d. (0,2 puntos) Explique la mayor ventaja de LSA con respecto a tf-idf desde el punto de vista de la eficiencia (en máximo dos líneas).

Semana 11

1. Se tiene el siguiente B-tree donde los nodos pueden contener como máximo 4 elementos:

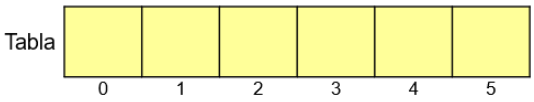


   a. (0,3 puntos) Dibuje el árbol resultante después de insertar los siguientes 5 elementos: **35**, **55**, **45**, **17**, **21**.

2. Se desea almacenar y buscar números enteros entre el 0 y el 99. Para eso se usará una tabla de hash de largo 6 con encadenamiento, usando la siguiente función de hash:

$$h(x) = \left\lfloor \frac{x}{4} \right\rfloor \bmod 6$$

   Donde $\lfloor \rfloor$ representa la parte entera[2] y $\bmod$ es el resto de la división entera[3], por ejemplo $h(46) = 5$.



   a. (0,4 puntos) Dibuje el estado de la tabla luego de agregar los siguientes 10 elementos: **10**, **24**, **37**, **40**, **46**, **56**, **72**, **76**, **84**, **92**.

   b. (0,2 puntos) Se desea buscar el elemento más cercano a una consulta, y por eficiencia se restringirá la búsqueda solo a los elementos asignados a la misma celda a la que sería asignada la consulta. Señale el elemento más cercano encontrado, entre los 10 elementos insertados, cuando la consulta es: **70**, **75** y **87**.

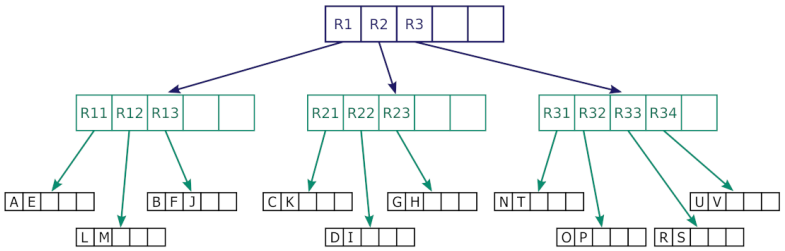
---

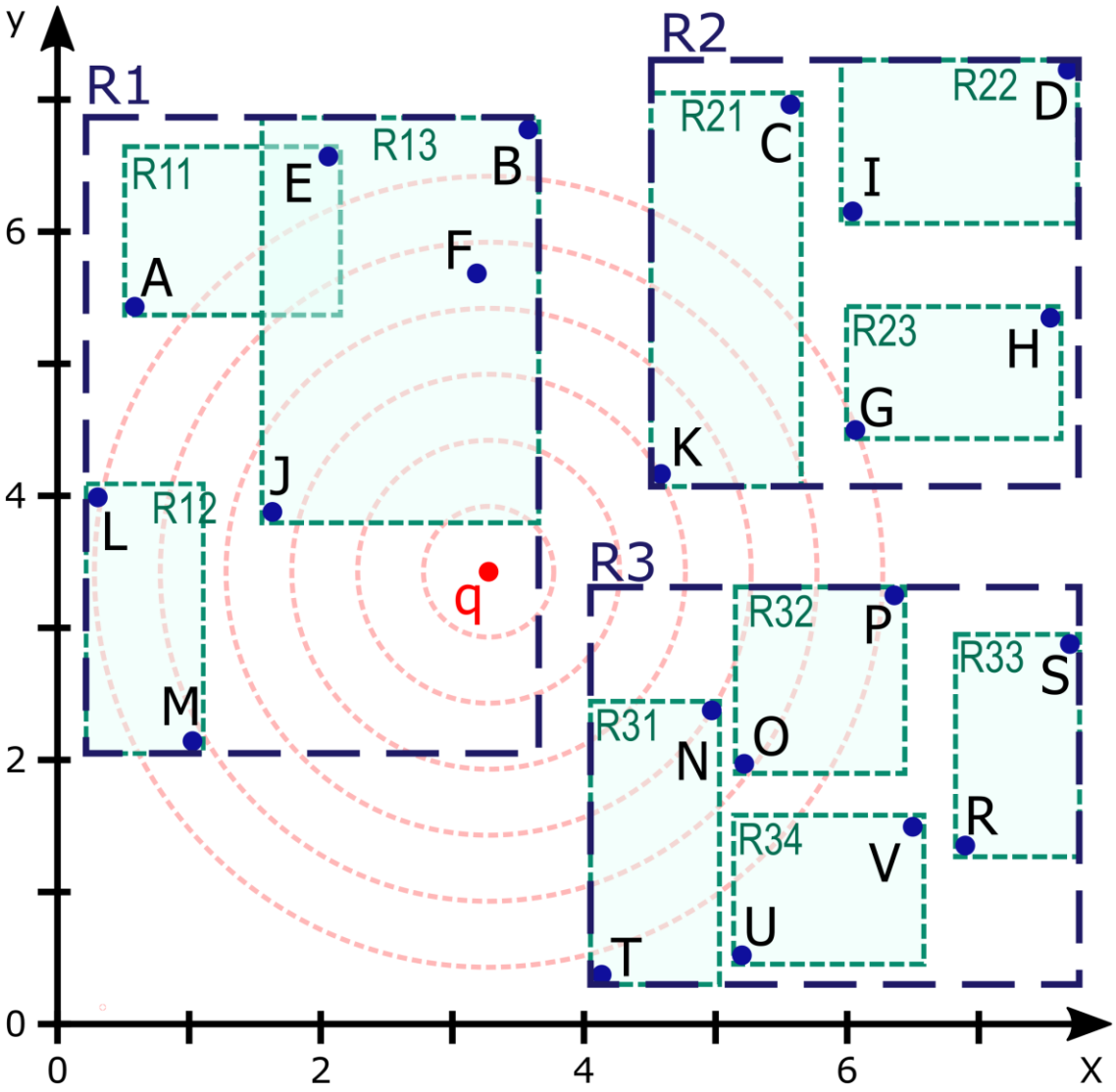
[2] Similar a la función math.floor() de python.

[3] Similar al operador % entre números de python.

Semana 12

1. Se tiene un conjunto de 21 vectores de dos dimensiones (de la A a la V). Los vectores fueron indexados por un R-Tree con la siguiente estructura:





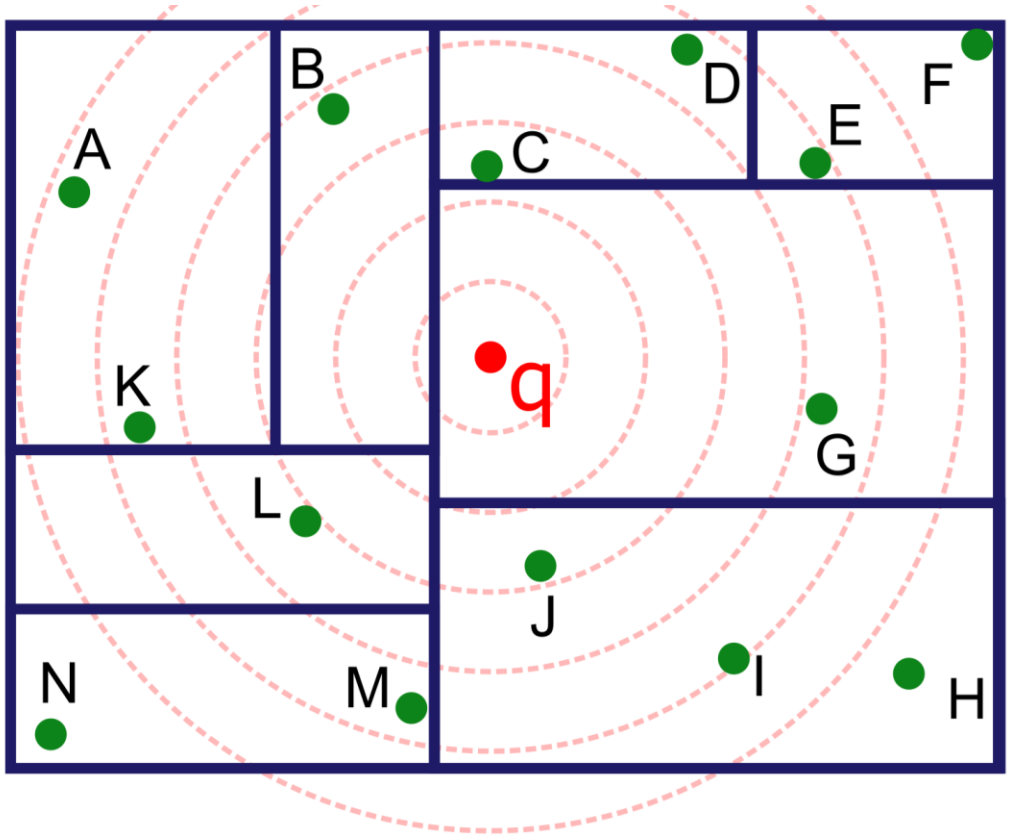
Se desea recuperar los **dos vecinos más cercanos (2-NN)** del objeto **q** de la figura anterior, según distancia Euclidiana. Usando el algoritmo de **búsqueda k-nn por prioridad**:

a. (0,5 puntos) Señale las regiones espaciales del árbol que son visitadas, el orden en que se visitan, los vectores que calculan su distancia a q y la evolución de los candidatos durante la búsqueda de los 2-NN.

b. (0,5 puntos) Señale el número de veces que se evaluó la distancia euclidiana y la distancia MINDIST al resolver la búsqueda 2-NN. Compare esta cantidad con la cantidad de evaluaciones de la función de distancia requeridas para el algoritmo linear scan.

Como ayuda visual, se muestran varios círculos concéntricos a **q**.

2. Se tiene un conjunto de 14 vectores de dos dimensiones (de la A a la N). Los vectores fueron indexados por un k-d tree. La siguiente figura muestra los vectores y los nodos hoja del árbol:



Se desea realizar una **búsqueda aproximada** del vecino más cercano a **q** según distancia Euclidiana. El parámetro de aproximación ***c*** es el número máximo de nodos hoja a visitar durante la búsqueda aproximada.

a. (0,6 puntos) Señale el nombre del vector que se encontrará como vecino más cercano cuando:

    i. La búsqueda aproximada del NN se restringe a ***c*** = 1.
    ii. La búsqueda aproximada del NN se restringe a ***c*** = 2.
    iii. La búsqueda aproximada del NN se restringe a ***c*** = 3.
    iv. La búsqueda aproximada del NN se restringe a ***c*** = 4.
    v. La búsqueda aproximada del NN se restringe a ***c*** = 5.
    vi. La búsqueda aproximada del NN se restringe a ***c*** = 6.

Como ayuda visual, se muestran varios círculos concéntricos a **q**.

Entrega

- Puede desarrollarlo en papel y enviar una foto (.jpg, .png), o puede desarrollarlo en formato digital en una planilla (.xlsx .ods), un documento (.docx) u otro formato exportado a .pdf.

- El plazo máximo de entrega es el **lunes 23 de junio de 2025** hasta las 23:59. Existirá una segunda fecha (por definir) para entregar su respuesta sin descuentos en la nota.

El control es *individual* y debe ser de su autoría, es decir, no pueden ser resueltos por otro estudiante, no se pueden copiar respuestas de Internet, no se permite usar ChatGPT ni similares. En caso de detectar copia o plagio se asignará nota 1.0 a las o los estudiantes involucrados.