K-D TREES, QUADTREES, OCTREES

K-D Trees – Introducción

- El problema en la recuperación asociativa se centra alrededor de un archivo F el cual es una colección de registros
- Cada registro R de F es una k-tupla ordenada (v_0, v_1, … , v_{k-1}) de valores los cuales son las claves, o atributos, del registro.
- Cada registro en el archivo es almacenado como un nodo en el árbol

K-D Trees - Definición

- Además de las k claves las cuales componen el registro, cada nodo contiene dos punteros, los cuales son nulos o puntos a otro nodo en árbol k-d
- Asociado con cada nodo un discriminador, el cual es un entero entre 0 y k-1 inclusive

K-D Trees - Terminología

- $K_0(P),\ldots,K_{k-1}(P)$
- LOSON(P) y HISON(P)
- DISC(P)
- Si j = DISC(P) ∧ Q en LOSON(P) → $K_j(Q) < K_j(P)$
- Si j = DISC(P) ∧ R en HISON(P) → $K_j(R) > K_j(P)$
- NEXTDISC(i) = (i+1)mod k,
- NEXTDISC(i) = DISC(LOSON(P))

K-D Trees - Terminología

- SUCCESSOR(P,Q) devuelve el LOSON o el HISON. Hagamos j el DISC(P); si $K_j(P) <> K_j(Q)$, entonces es claro por la propiedad de los arboles k-d cual hijo sucesor devolver
- $S_{j-}(P) = K_j(P)K_{j+1}(P)\ldots K_{k-1}(P)K_0(P)\ldots K_{j-1}(P)$
- Si $S_j(Q) < S_j(P)$, SUCCESSOR devuelve LOSON;
- Si $S_j(Q) > S_j(P)$, devuelve HISON.
- Si $S_j(Q) = S_j(P)$ SUCCESSOR devuelve un valor especial para indicar esa situación

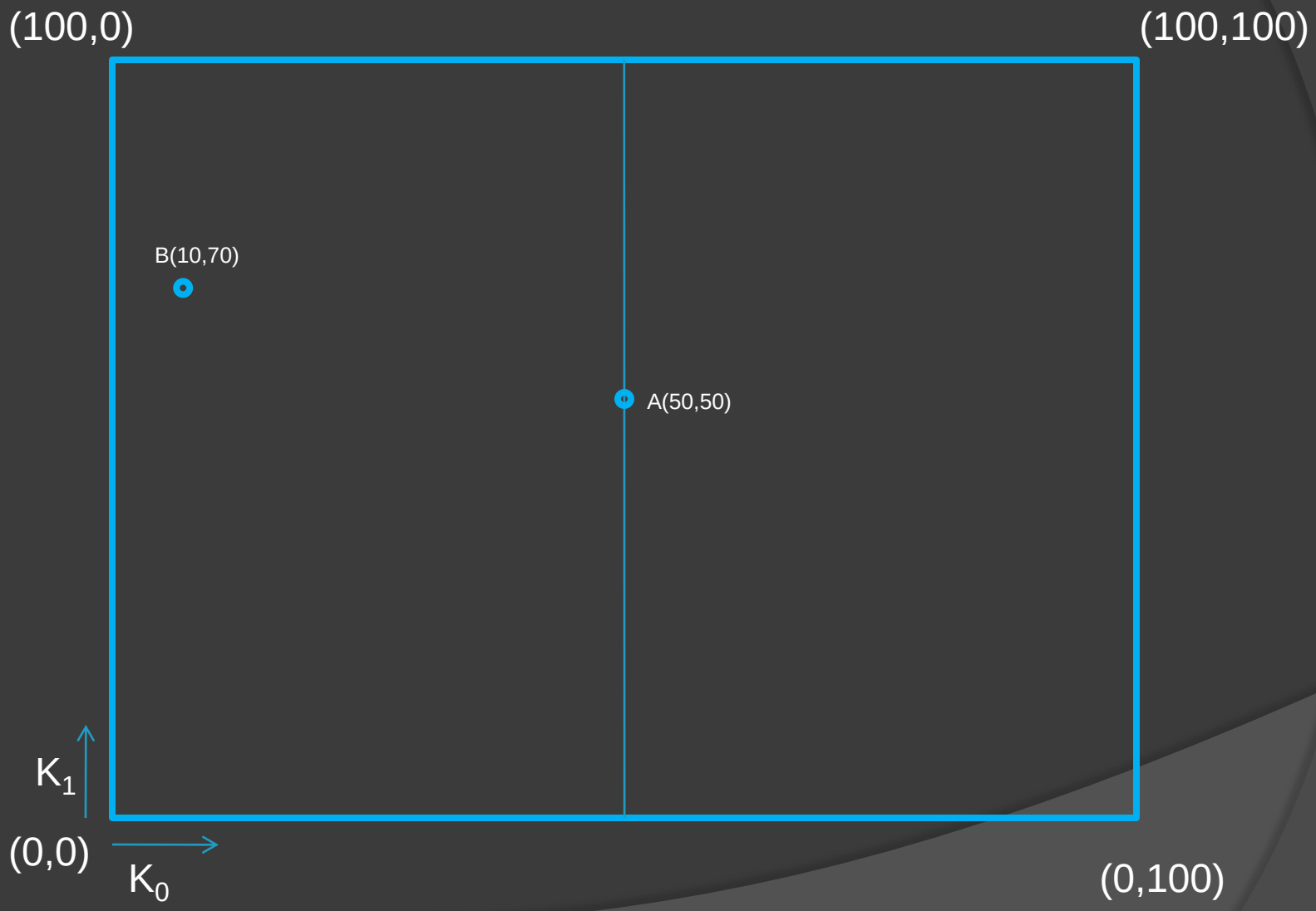
K-D Trees - Inserción

```
1   SI (ROOT = NIL ) entonces
                ROOT = P
    SI NO
                Q = ROOT
    FIN SI
2   SI ( Ki(P) = Ki(Q) ) entonces
        REGRESAR Q
    SI NO
        SON = SUCCESSOR(Q,P)
    FIN SI
    SI(SON(Q) = NIL) entonces
        IR A (4)
    FIN SI
3   Q = SON (Q)
    IR A (2)
4   SON(Q) = P
    HISON(P) = NIL
    LOSON(P) = NIL
    DISC(P) = NEXTDISC(DISC(Q))
    REGRESAR NIL
```

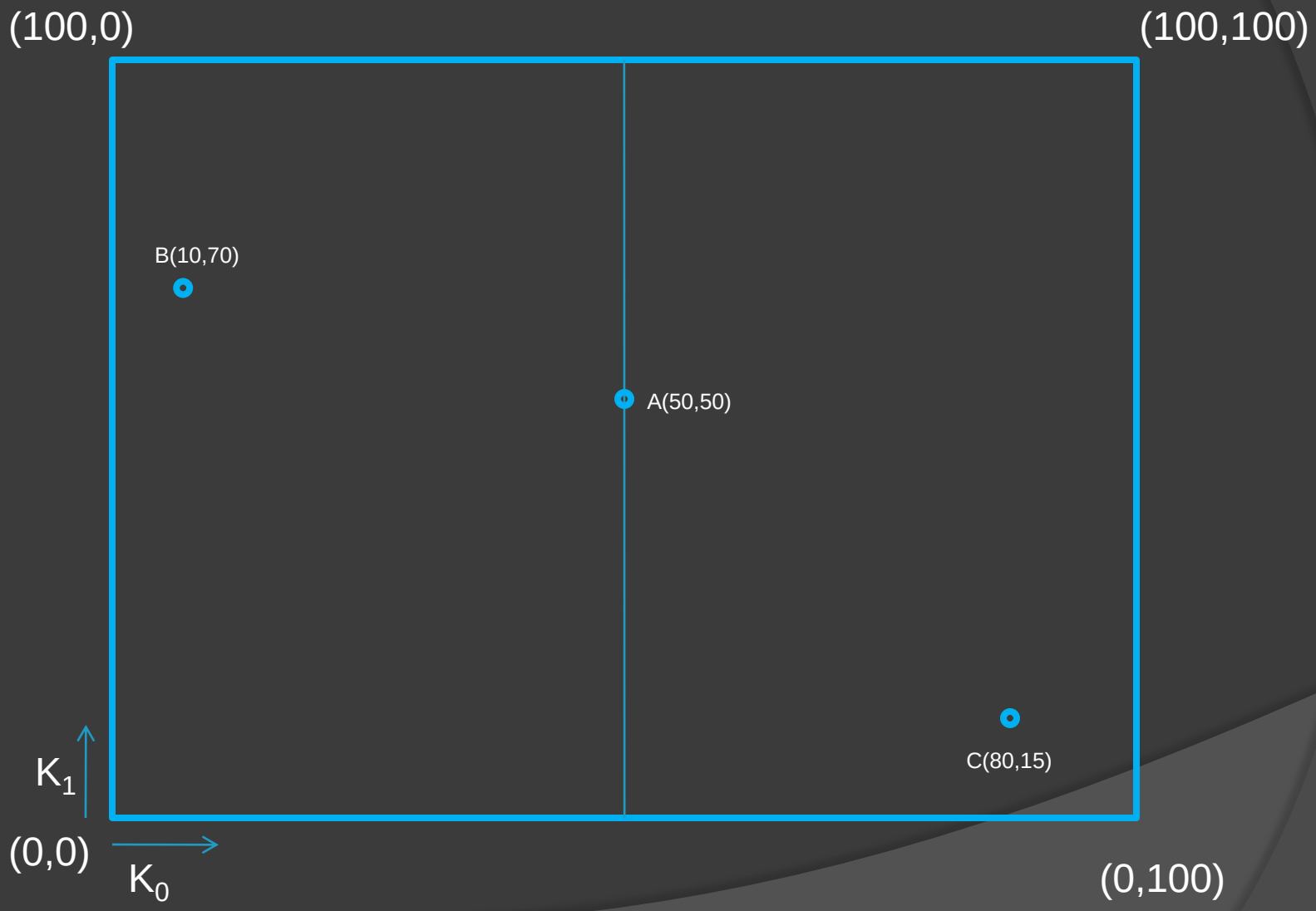
K-D Trees Representación Grafica

K-D Trees Representación Grafica

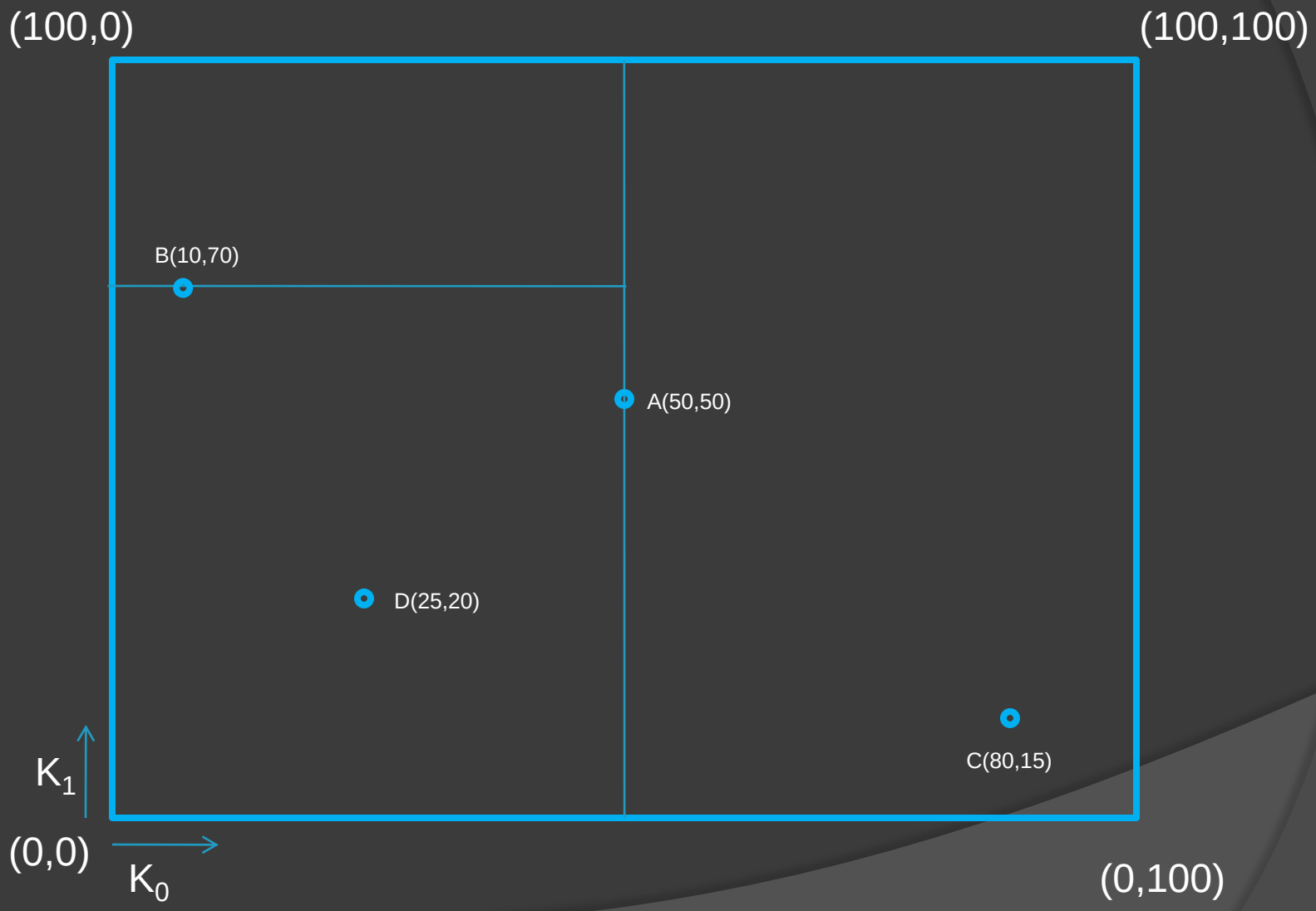
K-D Trees Representación Grafica



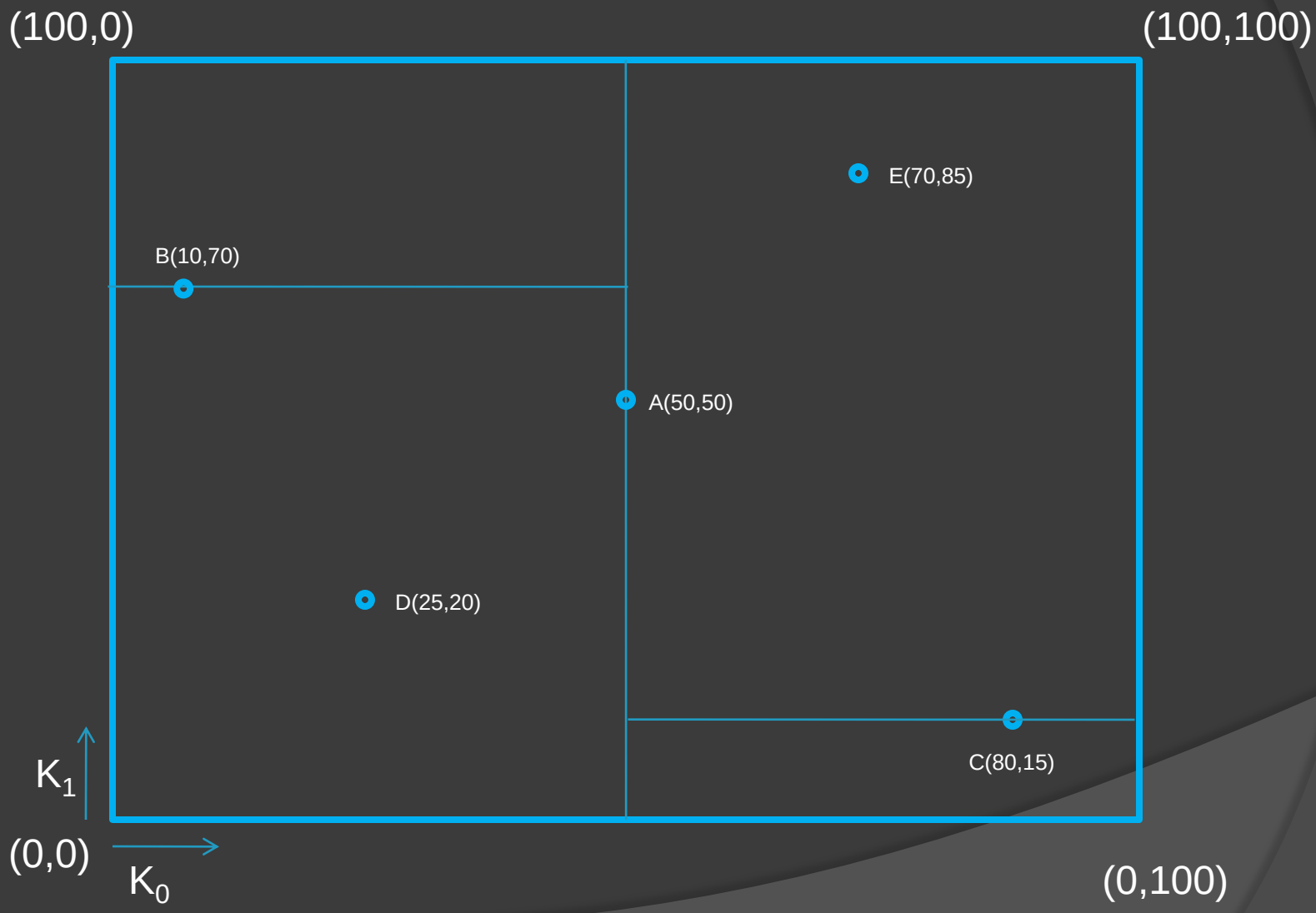
K-D Trees Representación Grafica



K-D Trees Representación Grafica



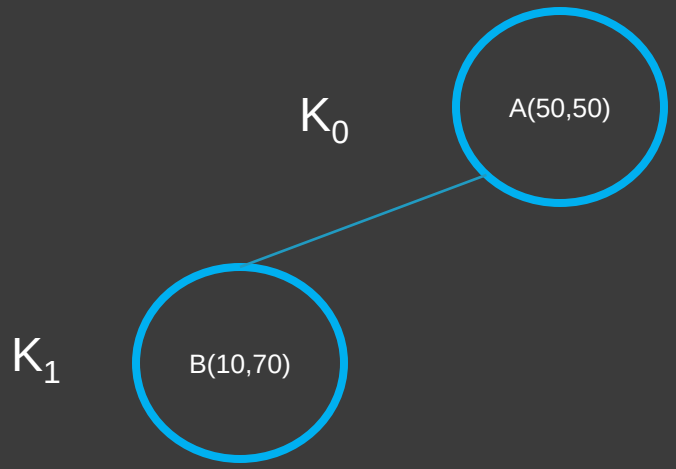
K-D Trees Representación Grafica



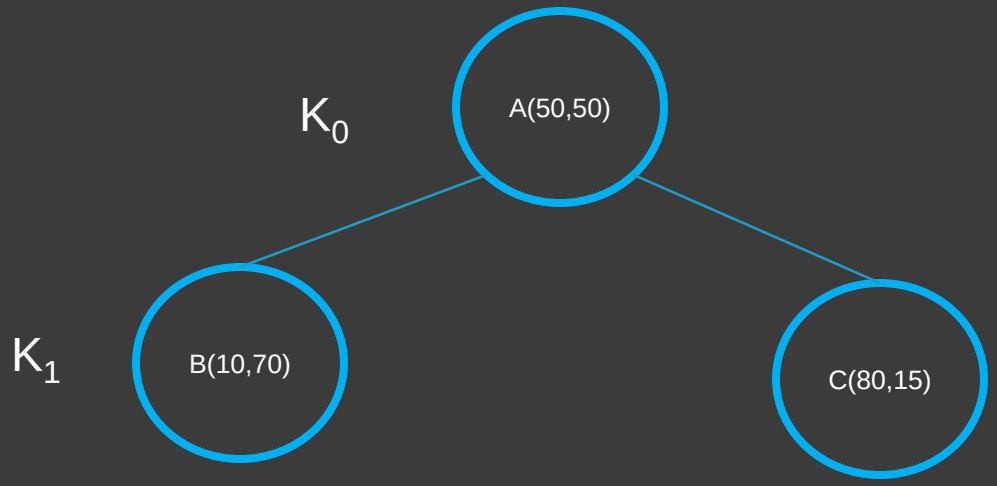
K-D Trees – Representación plana de grafos

K_0

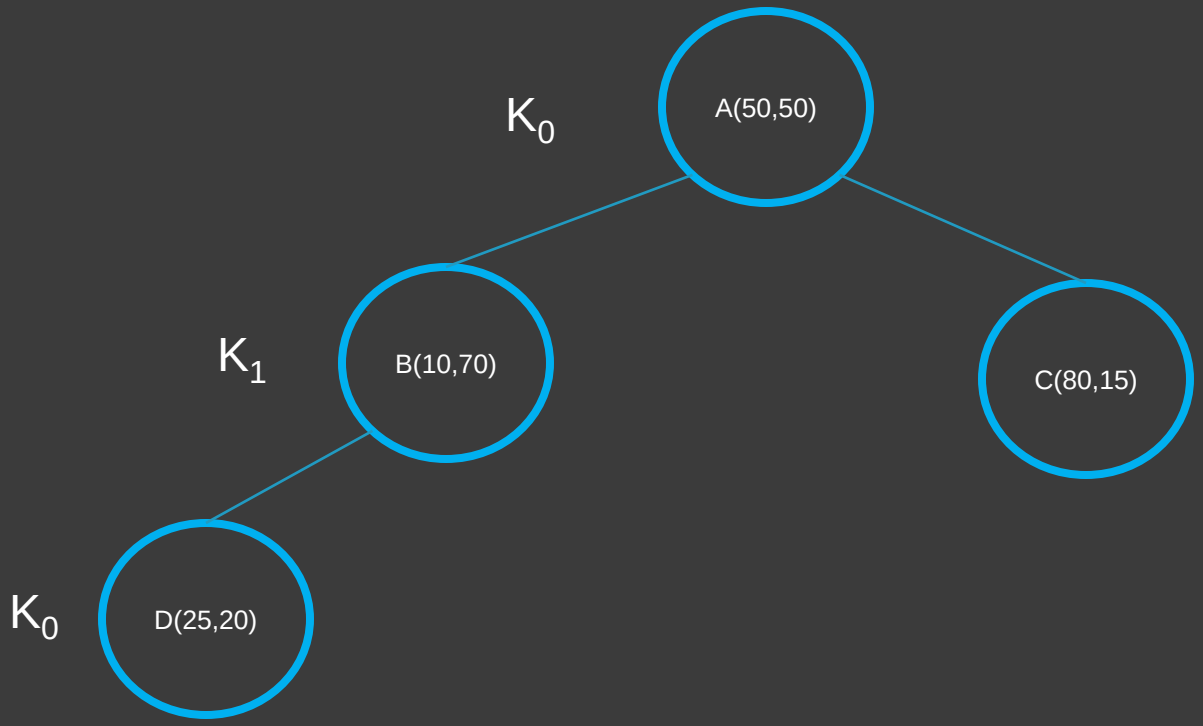
K-D Trees – Representación plana de grafos



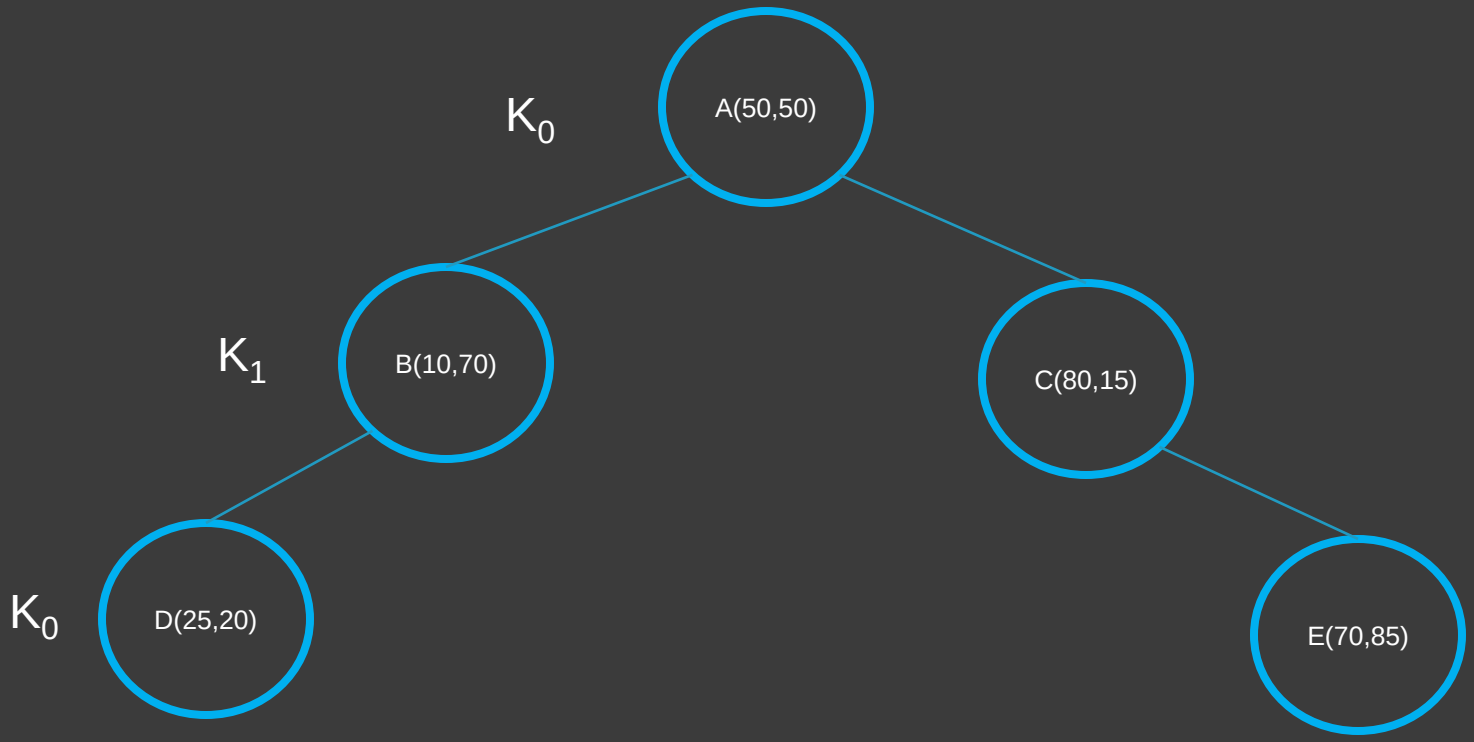
K-D Trees – Representación plana de grafos



K-D Trees – Representación plana de grafos



K-D Trees – Representación plana de grafos



K-D Trees - Consultas

Consultas Exactas

- La búsqueda procede hacia abajo en el árbol, yendo izquierda o derecha comparando con las claves de los registros deseados en el discriminador en el nodo. El numero de comparaciones para alcanzar una búsqueda exacta es O(lgN)

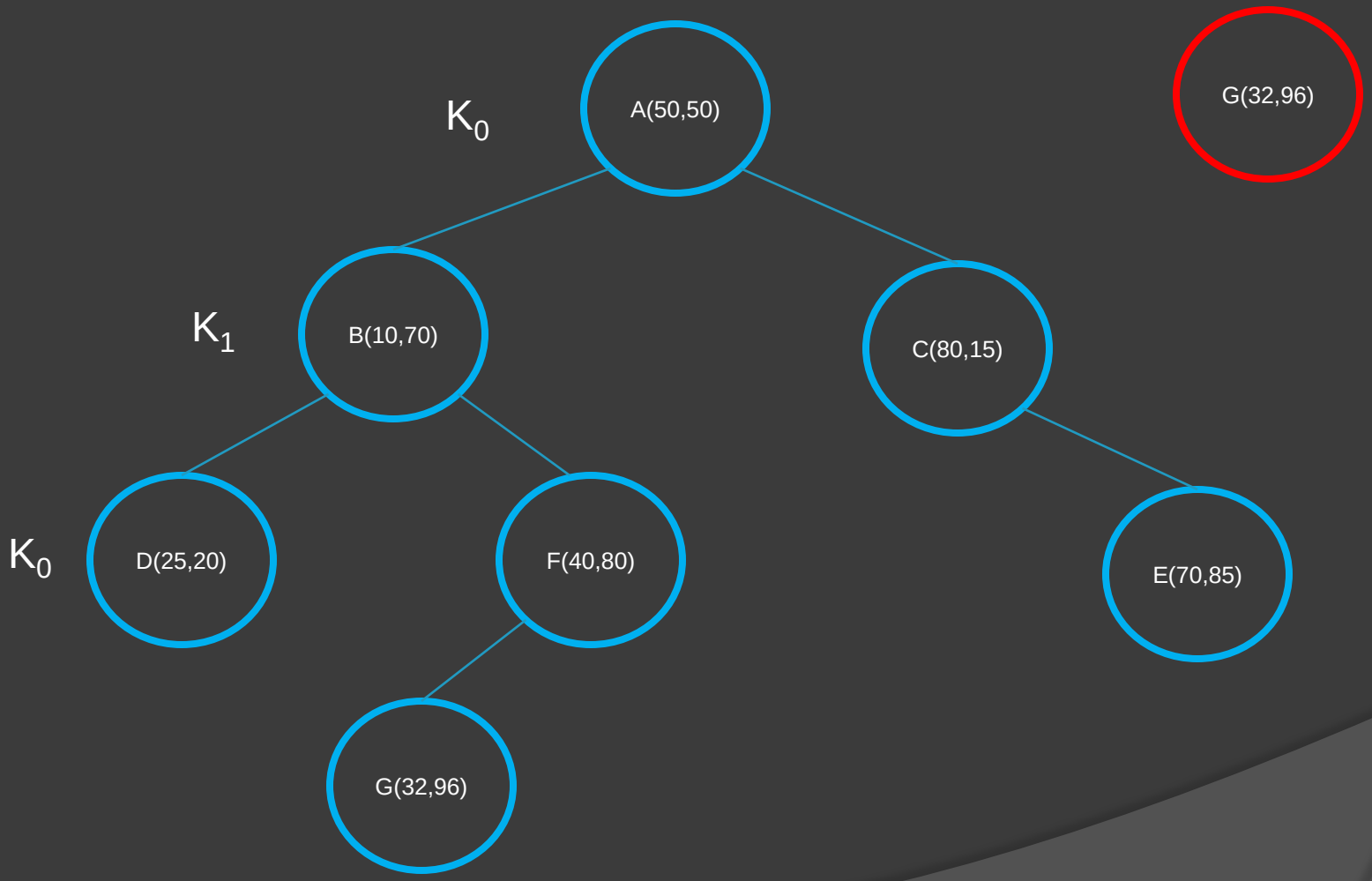
Consultas Parciales

- Cuando se visita un que discrimina por un valor j chequeamos si el valor de la j-esima clave esta especificado en la consulta, si lo está, entonces necesitamos solamente visitar uno de los nodos hijos. Si K_j no es una de las t claves especificadas, entonces debemos recursivamente buscar ambos hijos. El tiempo en un archivo de N registros es aproximadamente $tN^{1-t-/k}$.

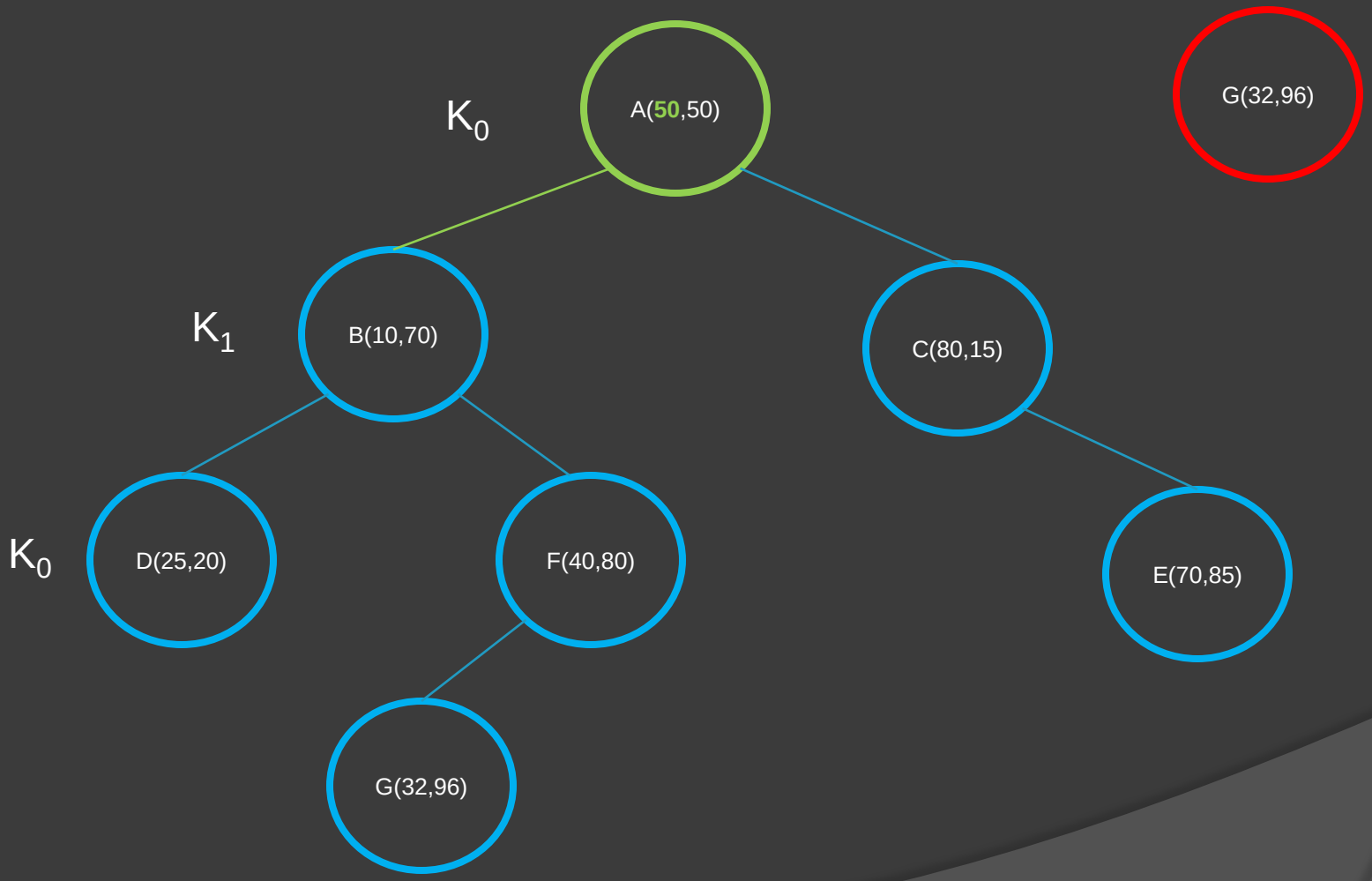
Consultas por Rangos

- Si el rango está completamente debajo de los valores entonces la búsqueda continua en el hijo izquierdo, si está completamente por encima entonces la búsqueda visita el hijo derecho, de otra manera ambos hijos son consultados recursivamente. El tiempo requerido para realizar una búsqueda por rangos nunca es más que $O(N^{1-1/k} + F)$, donde F es el número de puntos encontrados en el rango. En el caso promedio se reporta un tiempo esperado de O(lgN + F).

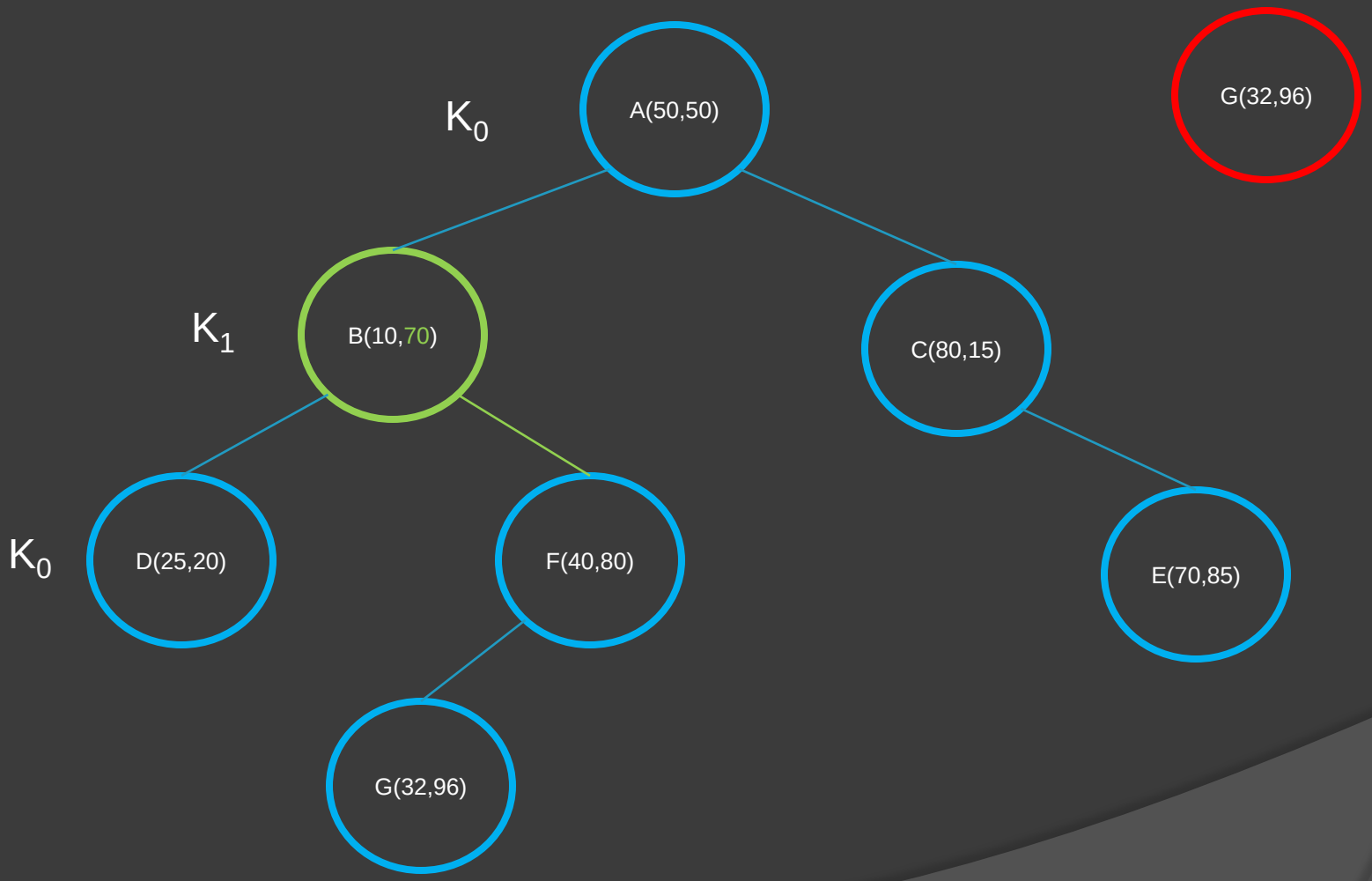
K-D Trees – Búsqueda Exacta



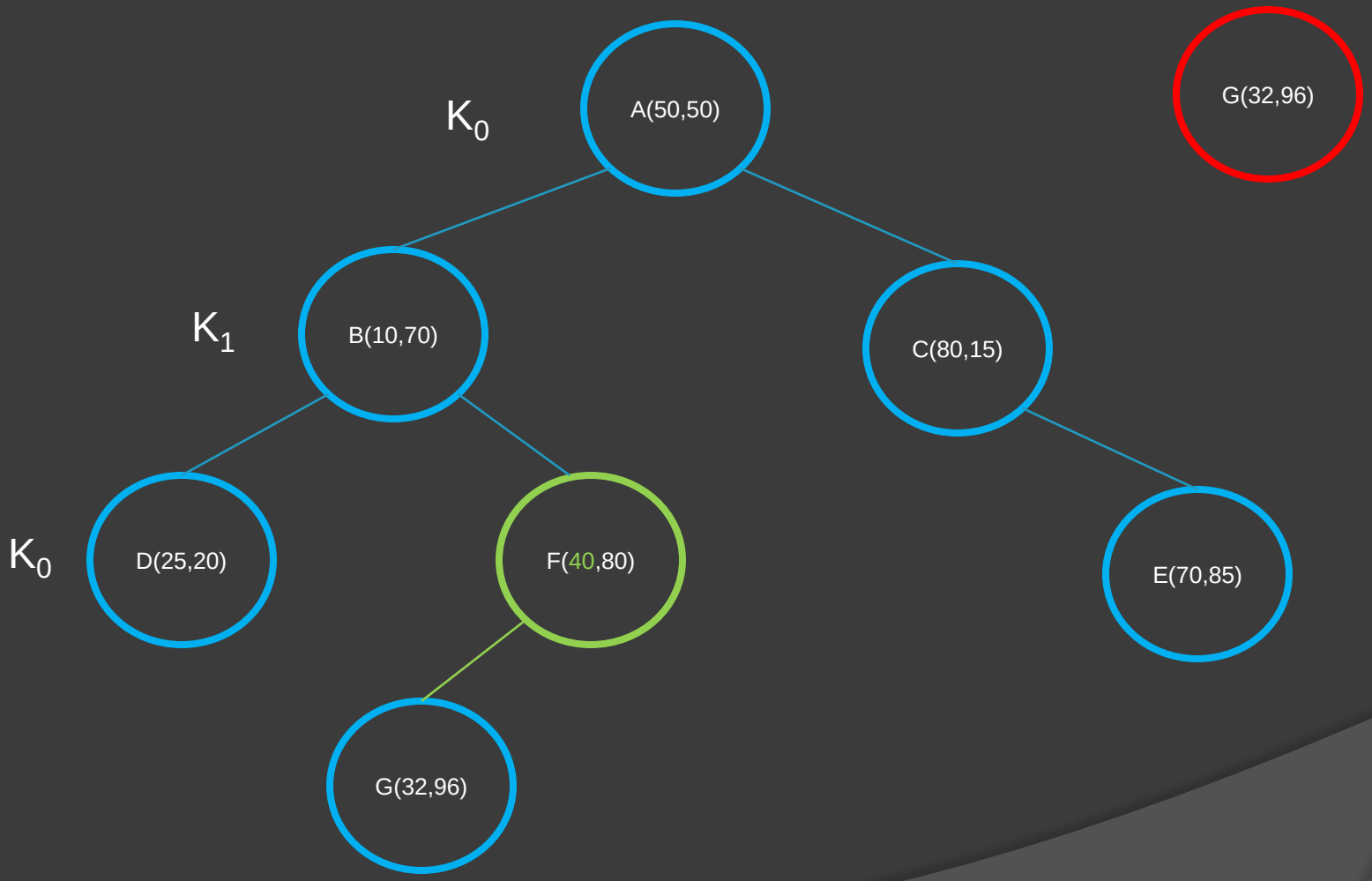
K-D Trees – Búsqueda Exacta



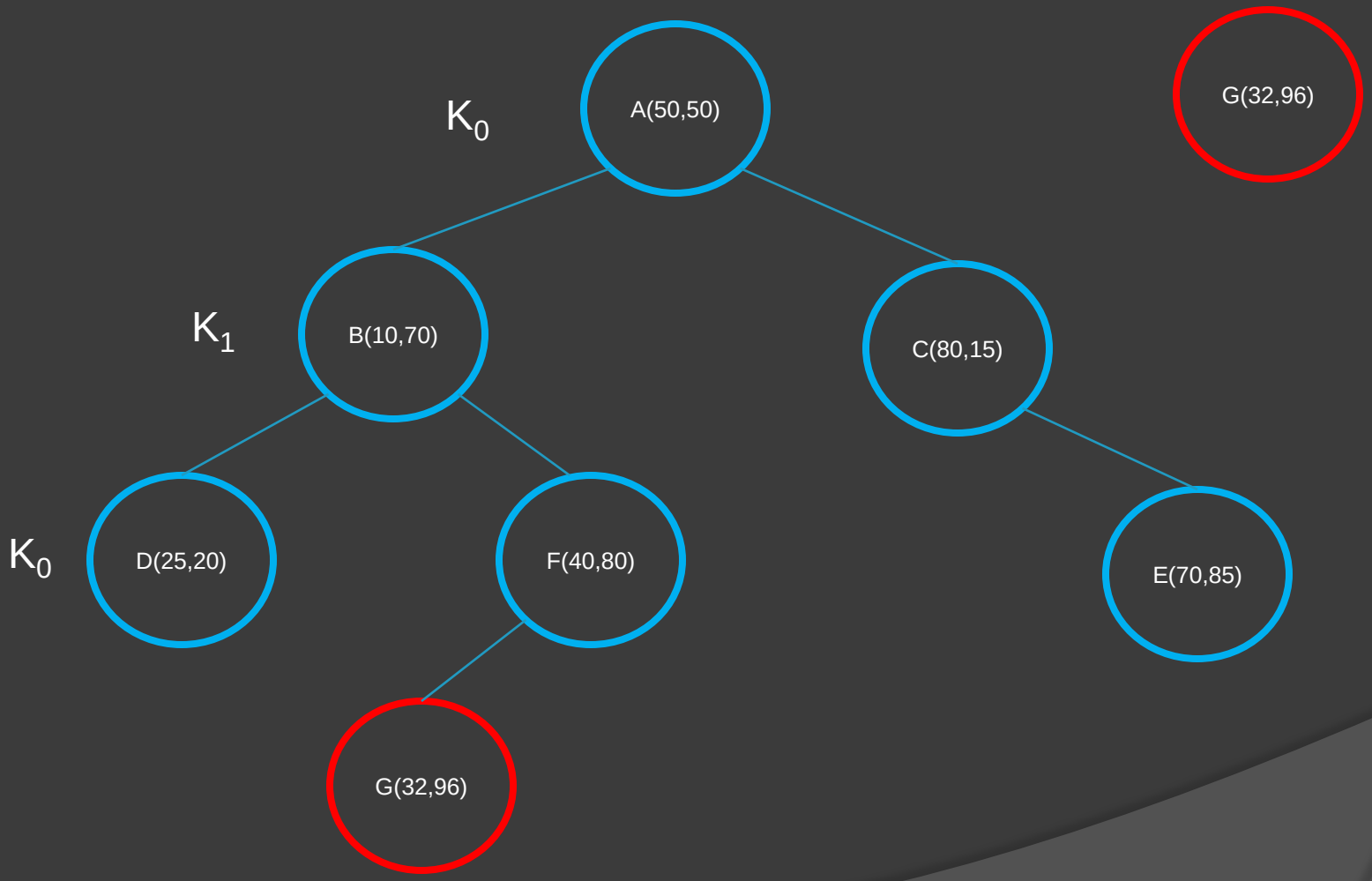
K-D Trees – Búsqueda Exacta



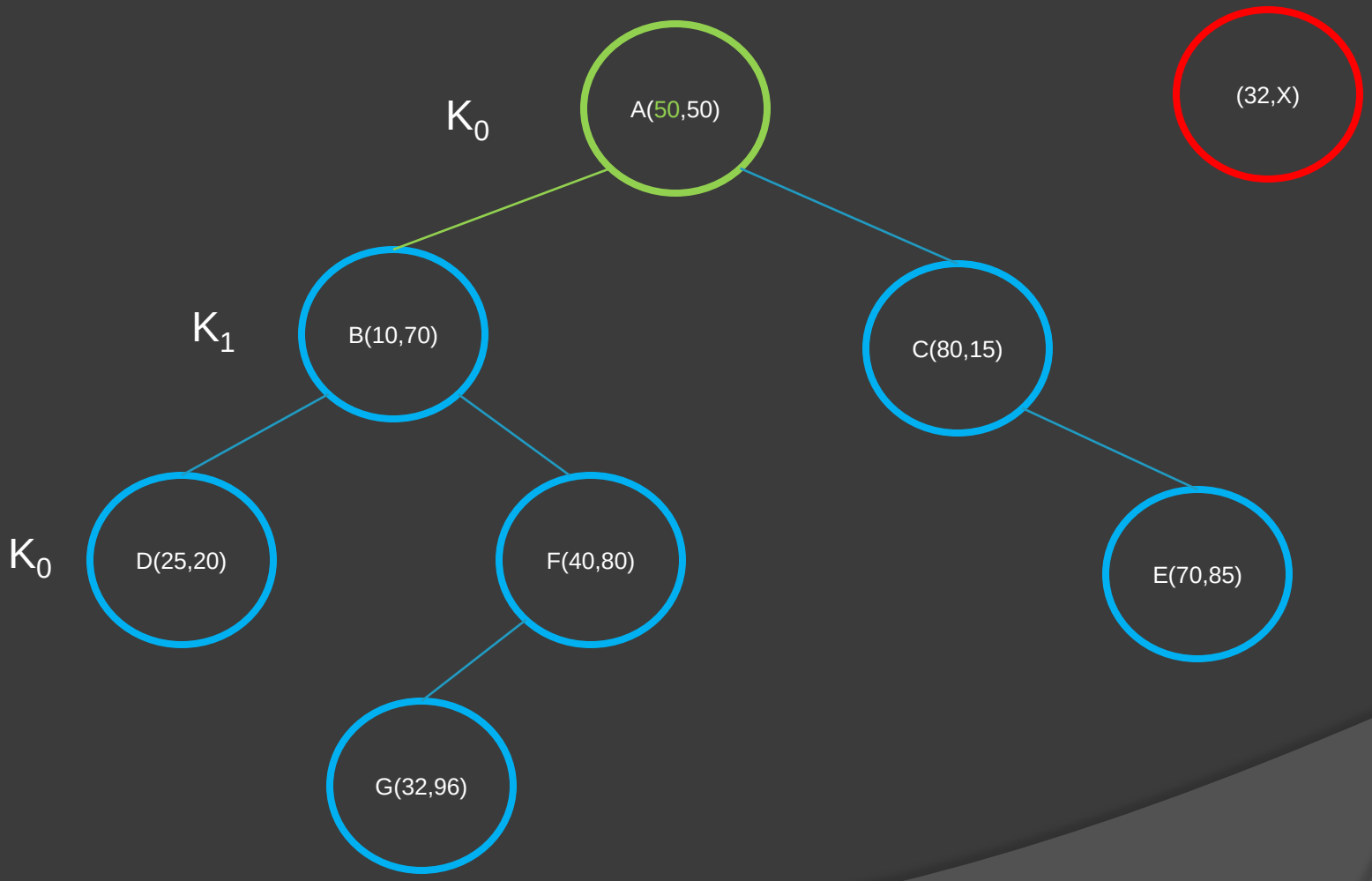
K-D Trees – Búsqueda Exacta



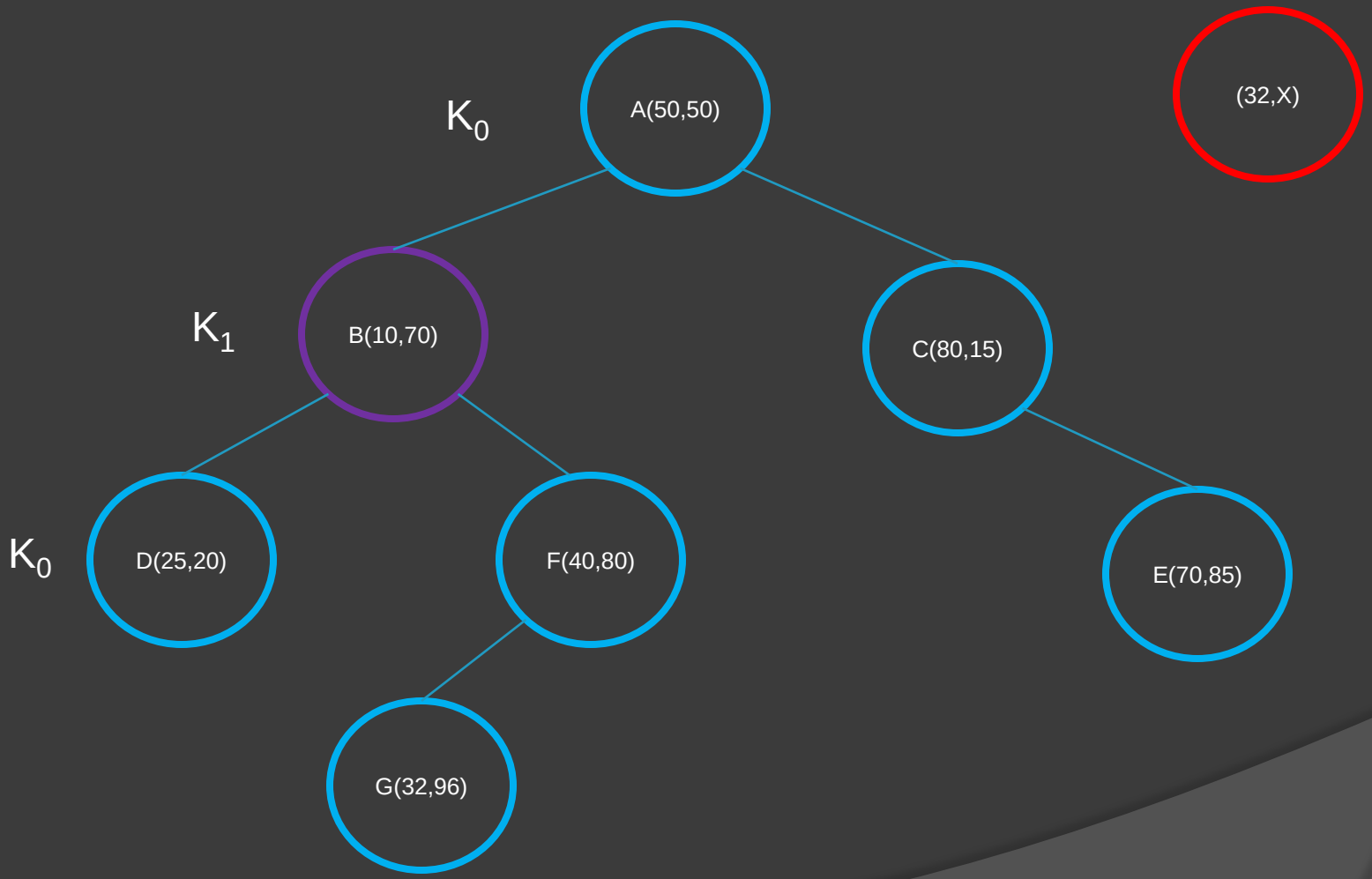
K-D Trees – Búsqueda Exacta



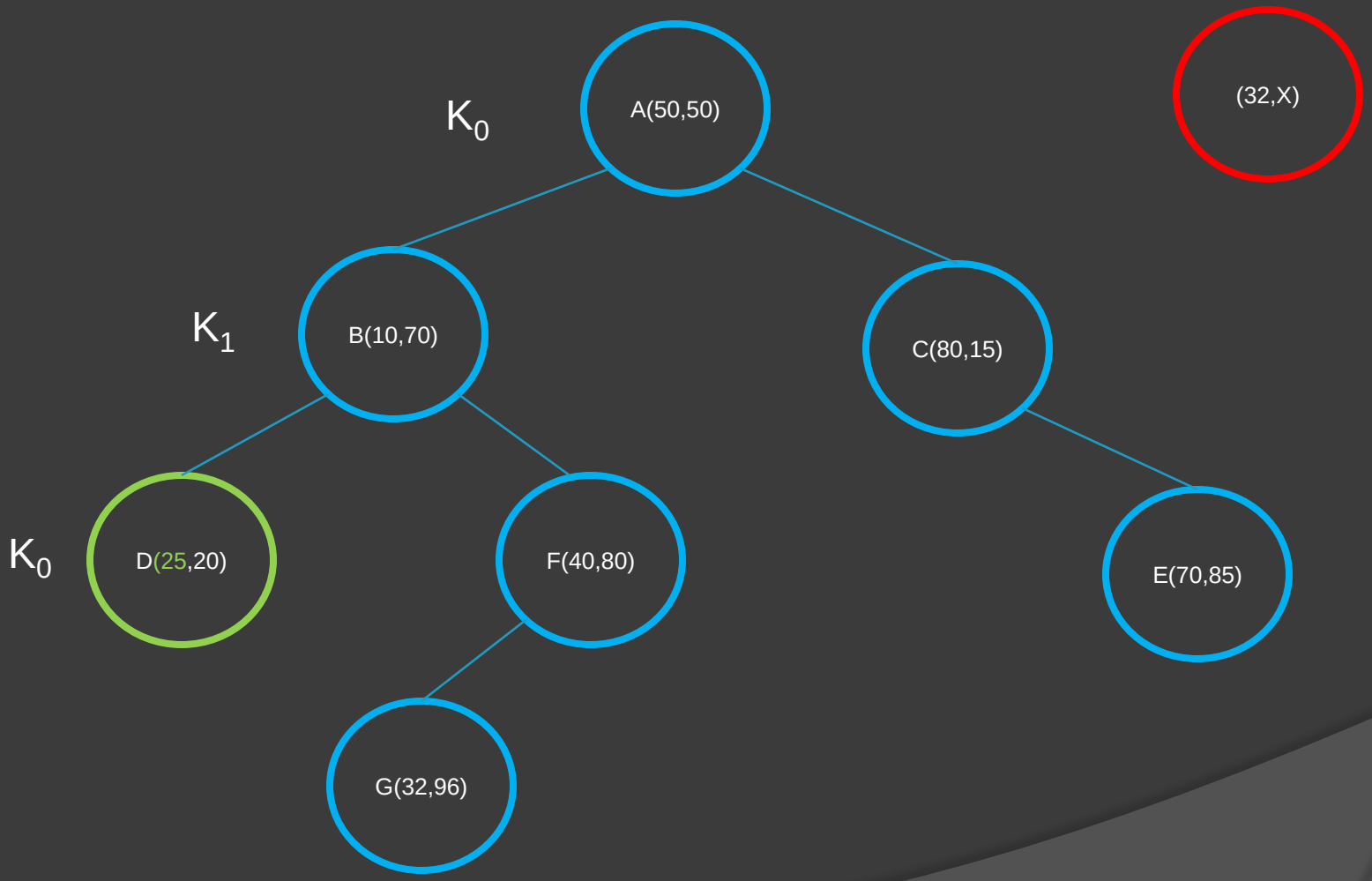
K-D Trees – Búsqueda Parcial



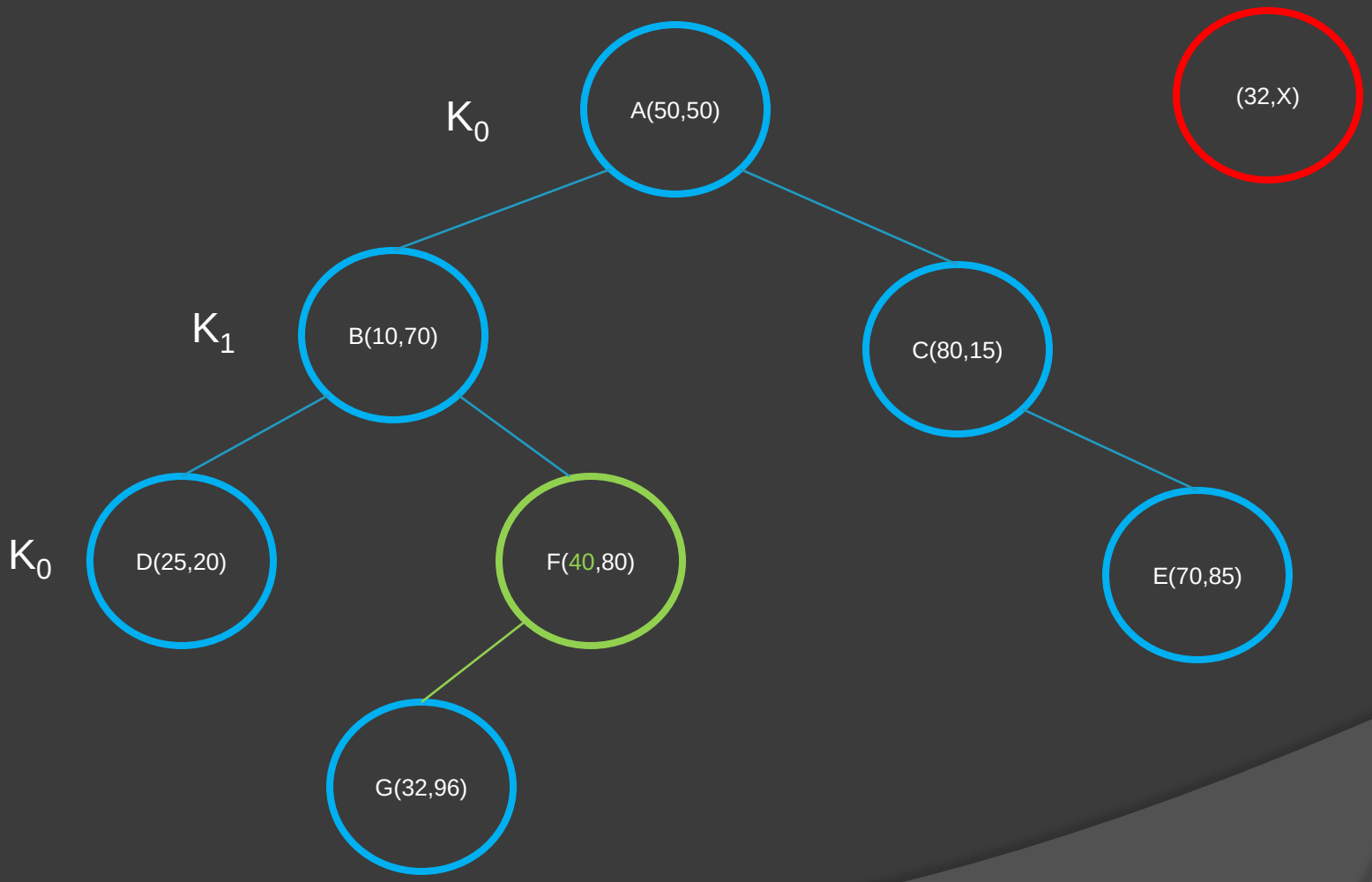
K-D Trees – Búsqueda Parcial



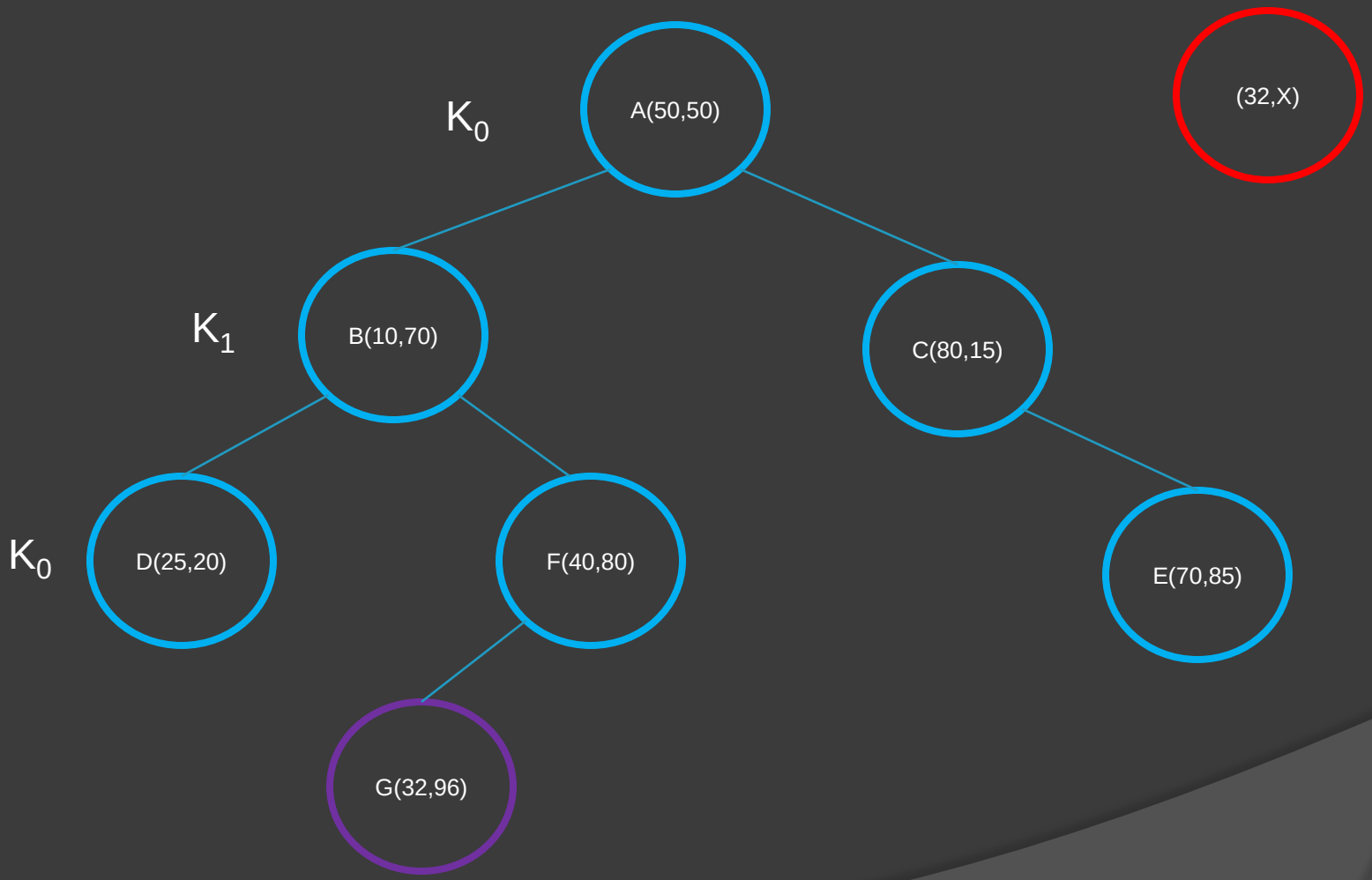
K-D Trees – Búsqueda Parcial



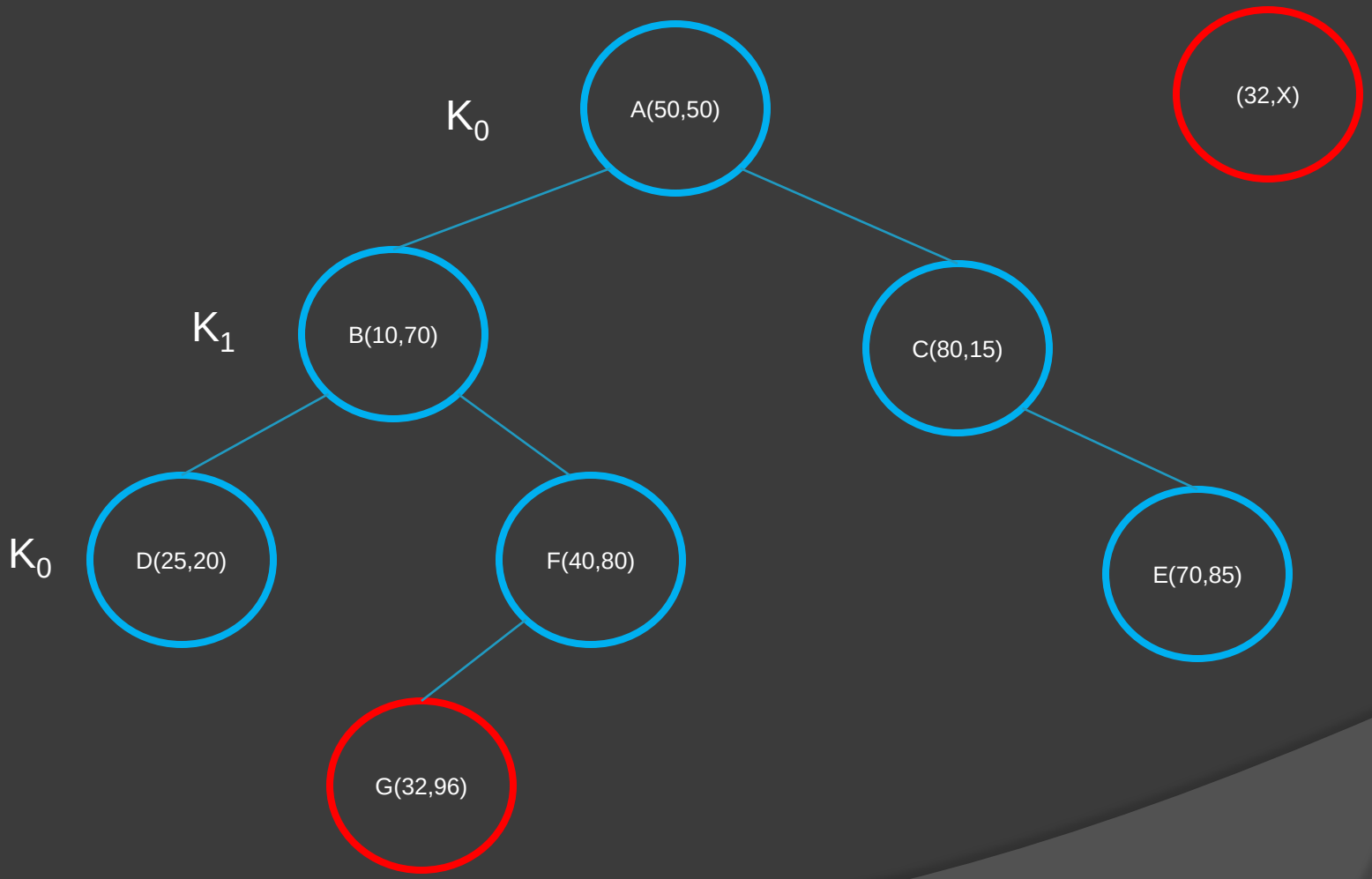
K-D Trees – Búsqueda Parcial



K-D Trees – Búsqueda Parcial



K-D Trees – Búsqueda Parcial



K-D Trees - Eliminación

1 SI (HISON(P) = NIL ∧ LOSON(P) = NIL) entonces
REGRESO NIL
SI NO
J = DISC(P)
FIN SI

2 SI HISON(P) = NIL entonces
IR A (4)
FIN SI

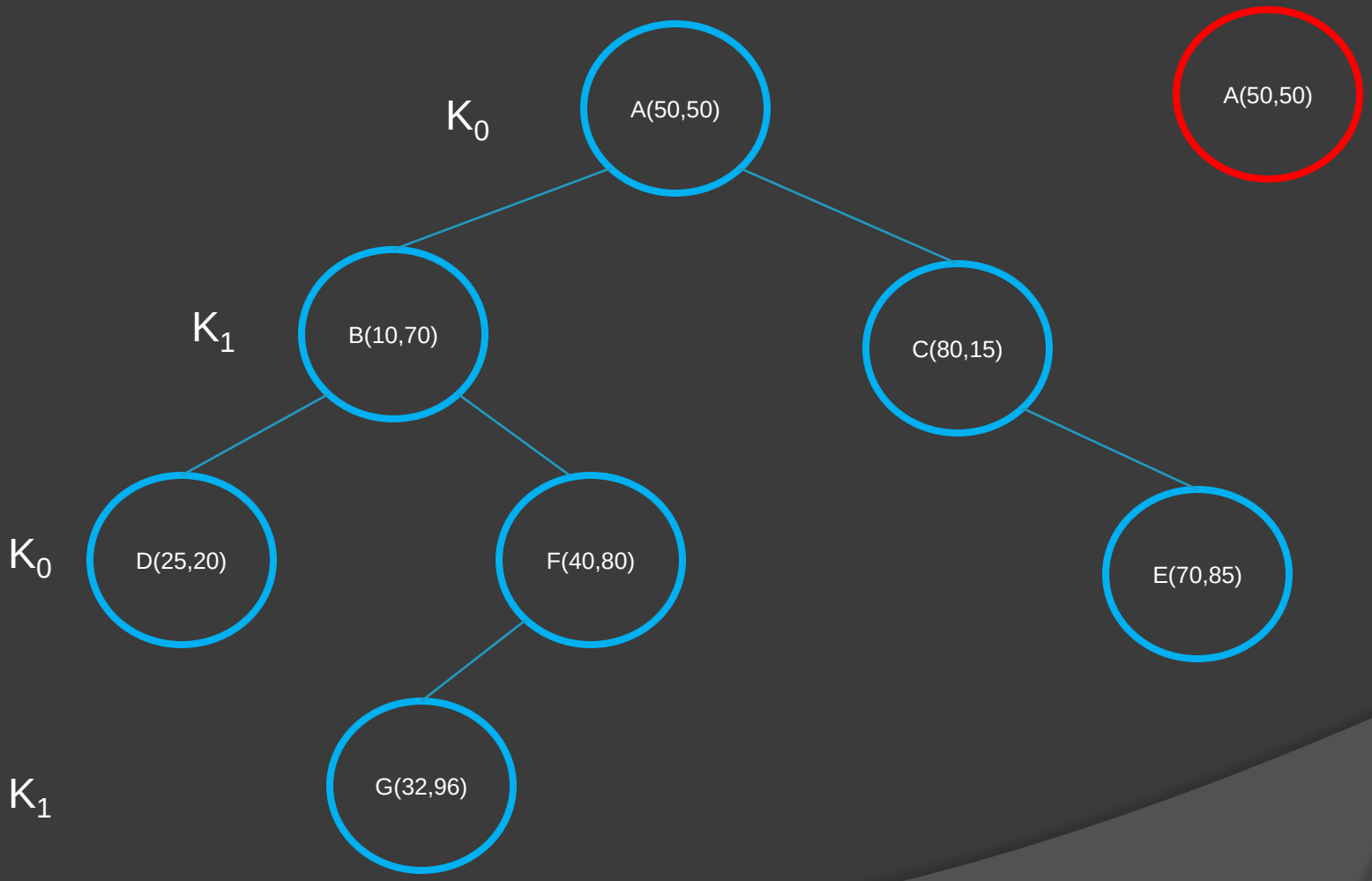
3 Q = Nodo J-mínimo en HISON(P)
QFATHER = padre de Q
QSON = cual hijo de Q lo va a ser de QFATHER
(Puede ser HISON o LOSON , QSON(QFATHER) = Q)
IR A (5)

4 Q = Nodo J-máximo en LOSON(P)
QFATHER = el padre de Q
QSON = cual hijo de Q es de QFATHER

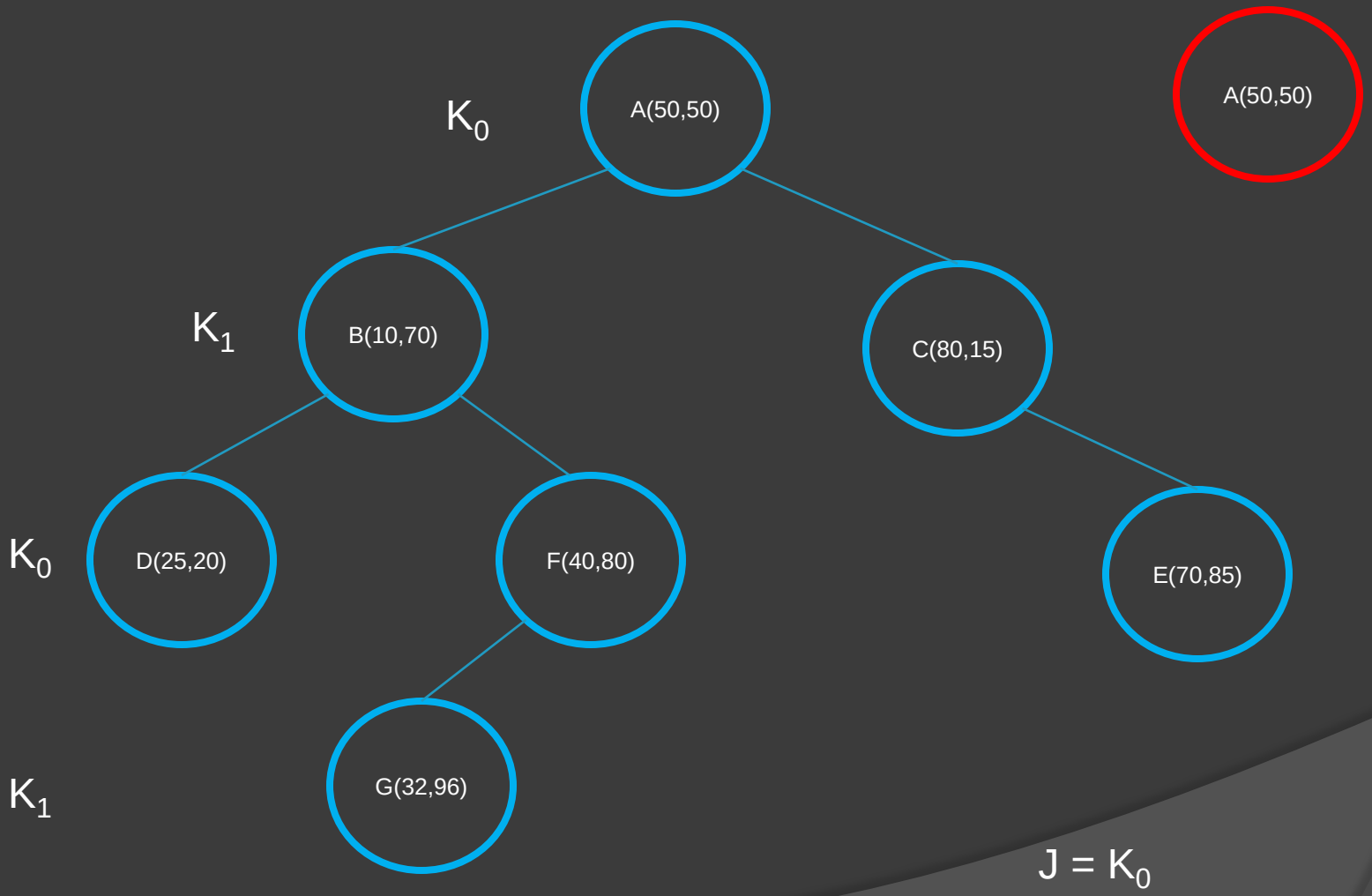
5 QSON(QFATHER) = DELETE(Q)
(Recursivamente libera Q para que pueda convertirse en la nueva raíz)

6 DISC(Q) = DISC(P)
HISON(Q) = HISON(P)
LOSON(Q) = LOSON(P)
REGRESAR Q

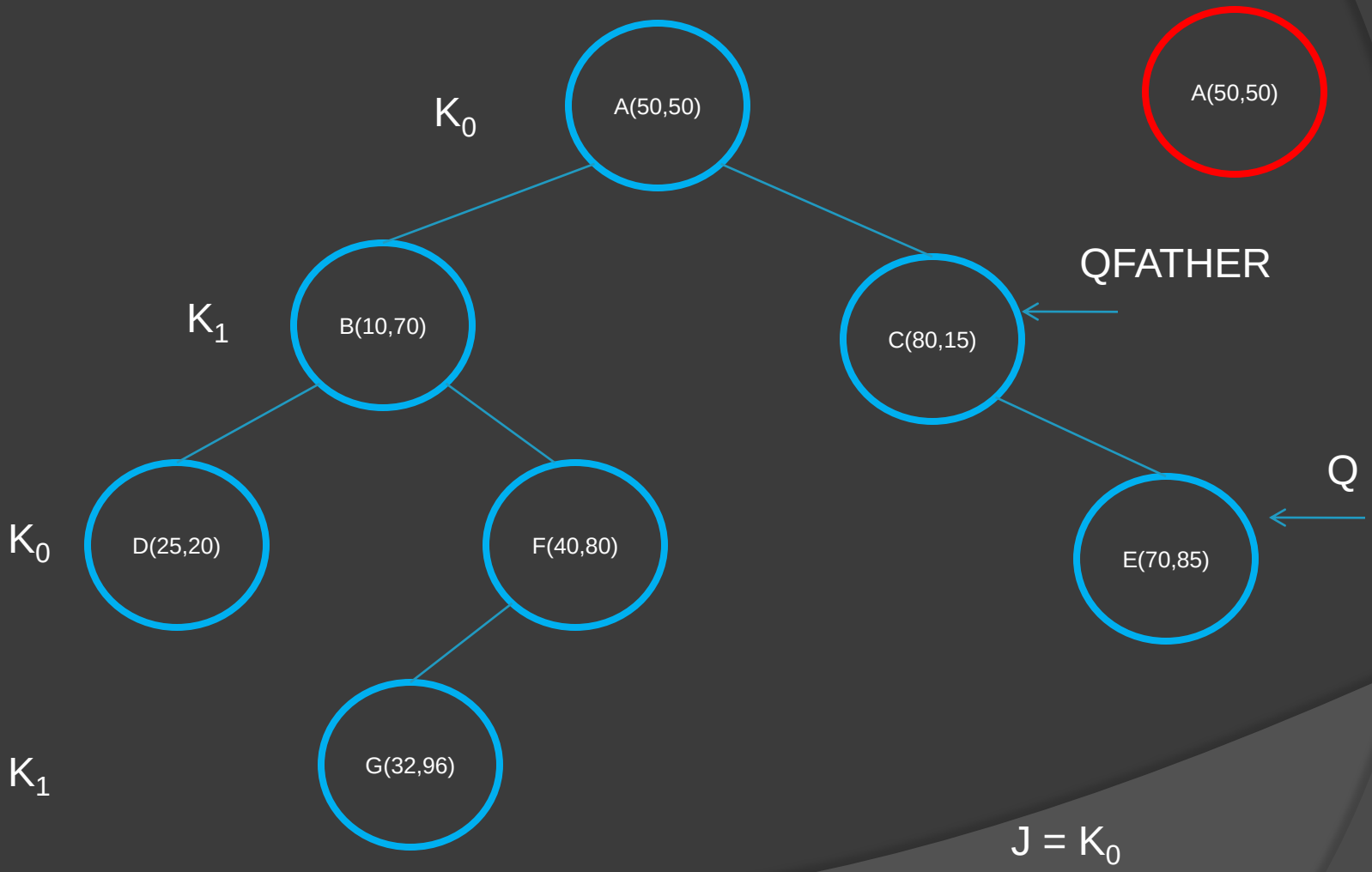
K-D Trees – Eliminación



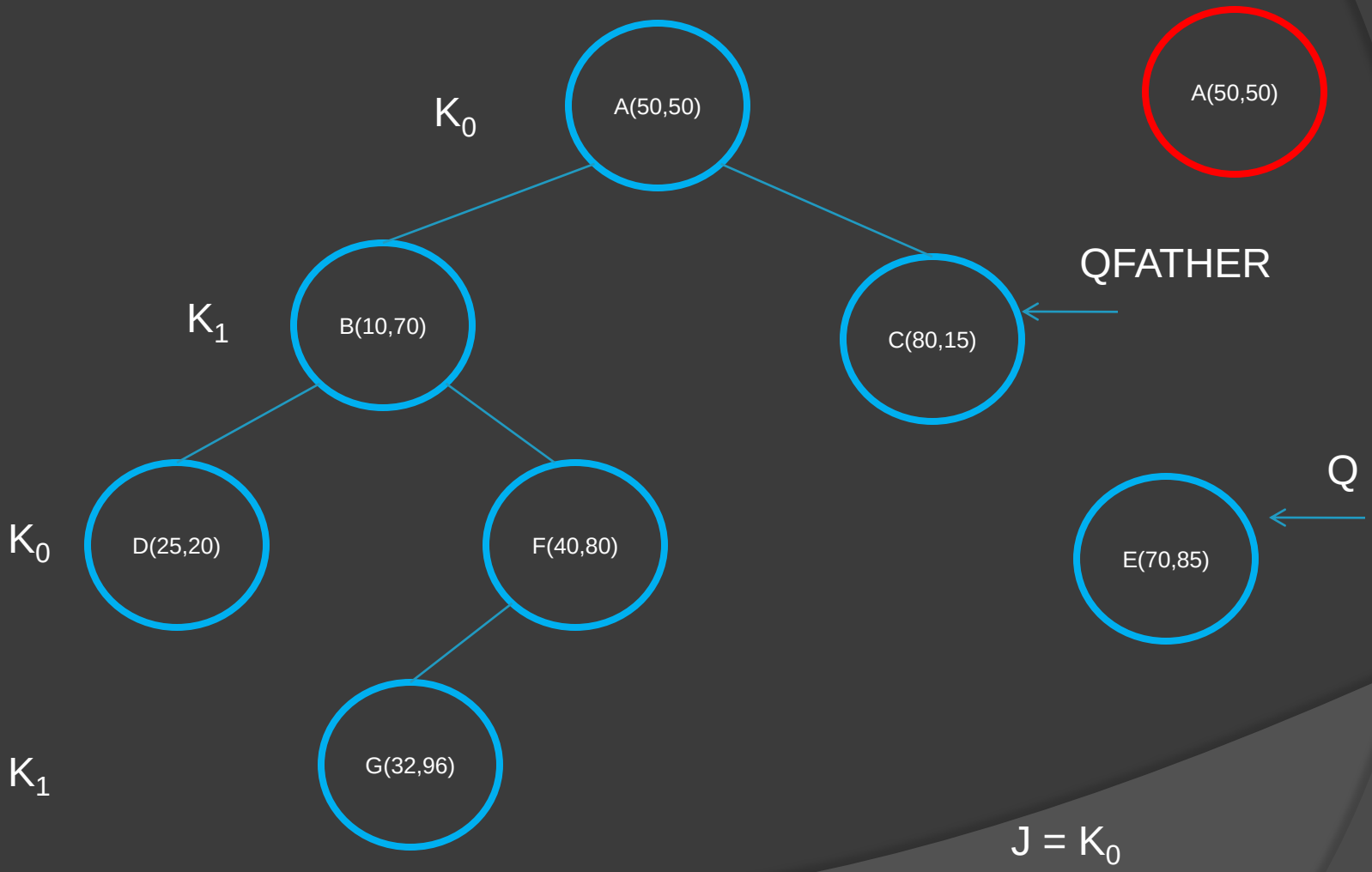
K-D Trees – Eliminación



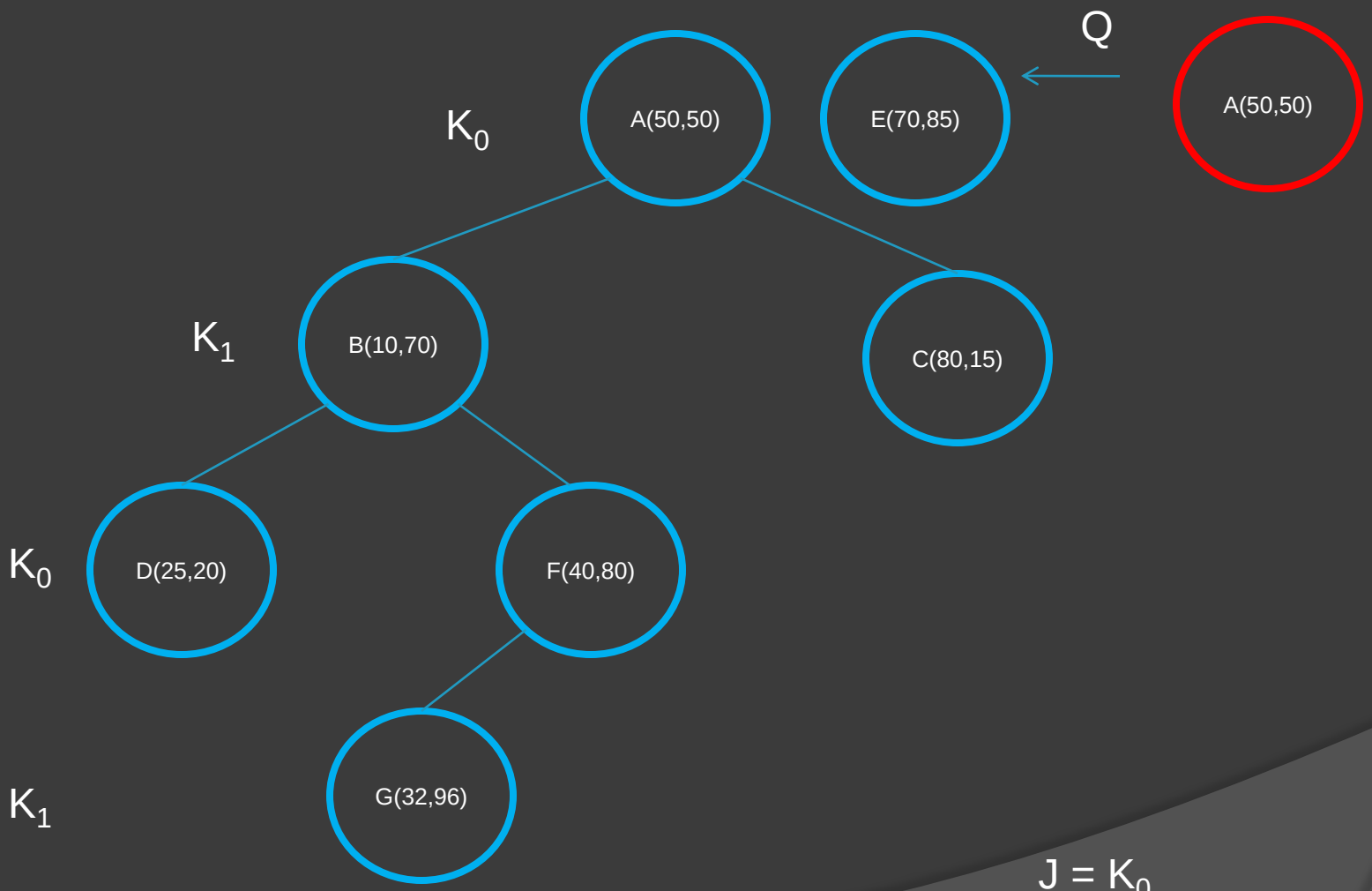
K-D Trees – Eliminación



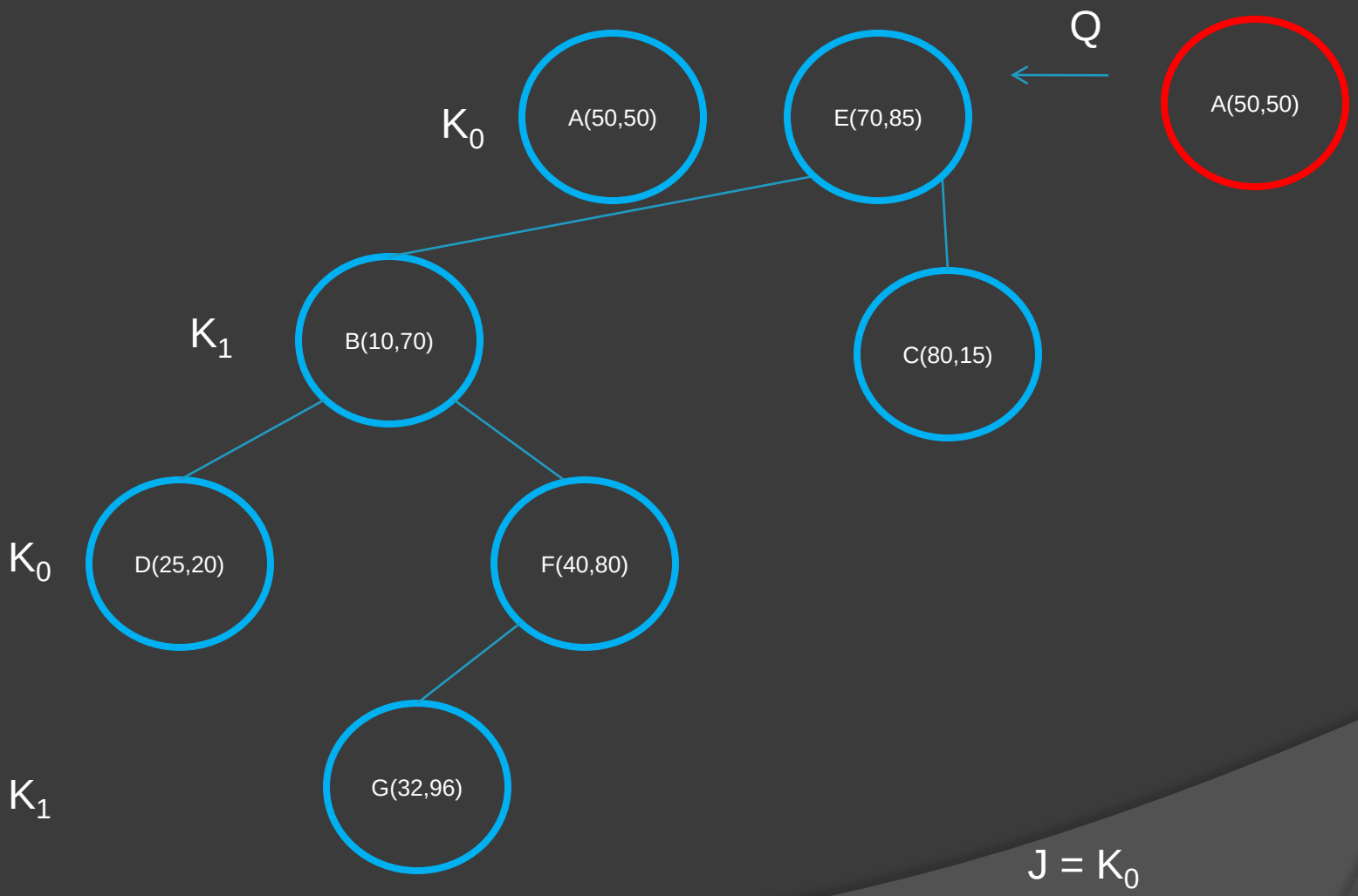
K-D Trees – Eliminación



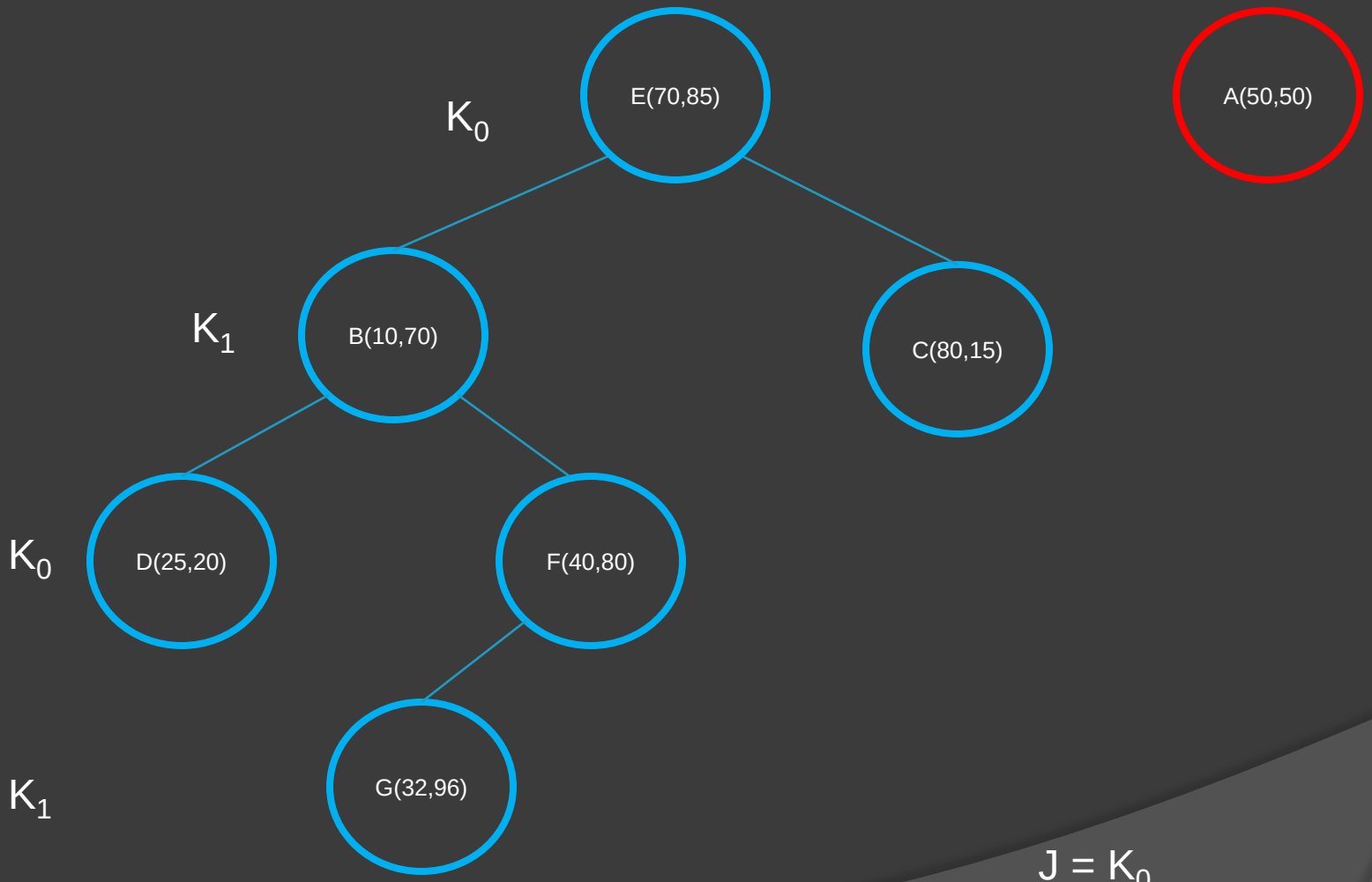
K-D Trees – Eliminación



K-D Trees – Eliminación



K-D Trees – Eliminación



Quadtrees - Introducción

- Describe una clase de estructuras de datos jerárquica cuyas propiedades comunes son que están basados en los principios de descomposición recursiva del espacio.

Diferencias

- El tipo de información que representa
- El principio que guía el proceso de descomposición
- La resolución (variable o no)

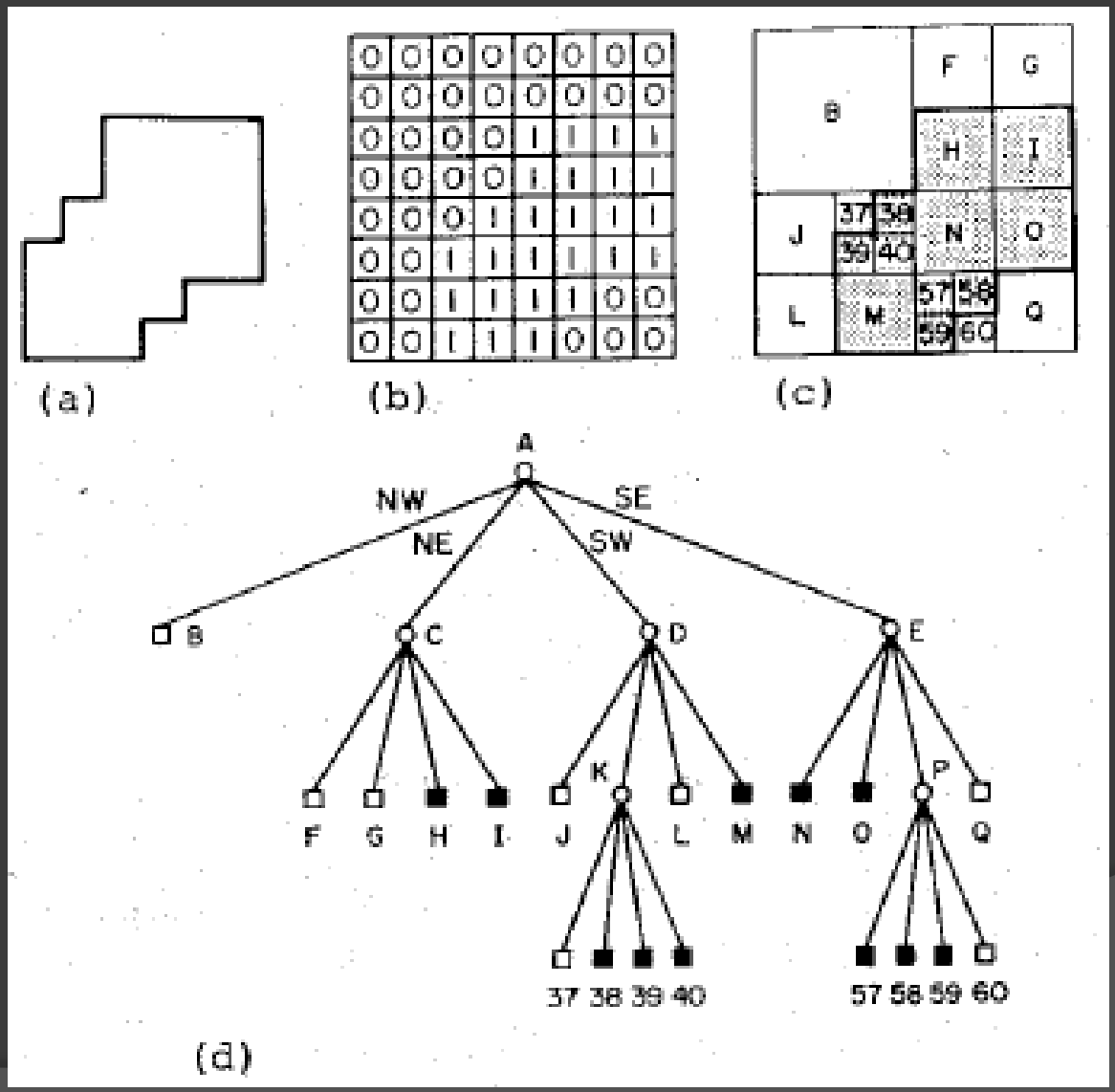
Quadtrees – Definición

- Son representados por un árbol de grado 4 en el cual la raíz representa un bloque y los cuatro hijos representan en orden los cuadrantes NW, NE, SW y SE.
- Registro contiene seis campos. Los primeros 5 campos contienen punteros a los nodos padre y sus cuatro hijos. El sexto campo, NODETYPE, describe el contenido del bloque de la imagen el cual el nodo representa.

Quadtrees - Terminología

- ADJ(B,I) es cierto si y solo si el cuadrante I es adyacente al límite B.
- COMMONSIDE(Q1,Q2) indica el límite del bloque contiene los cuadrantes Q1 y Q2.
- OPQUAD(Q) es el cuadrante el cual no comparte un bloque limite.

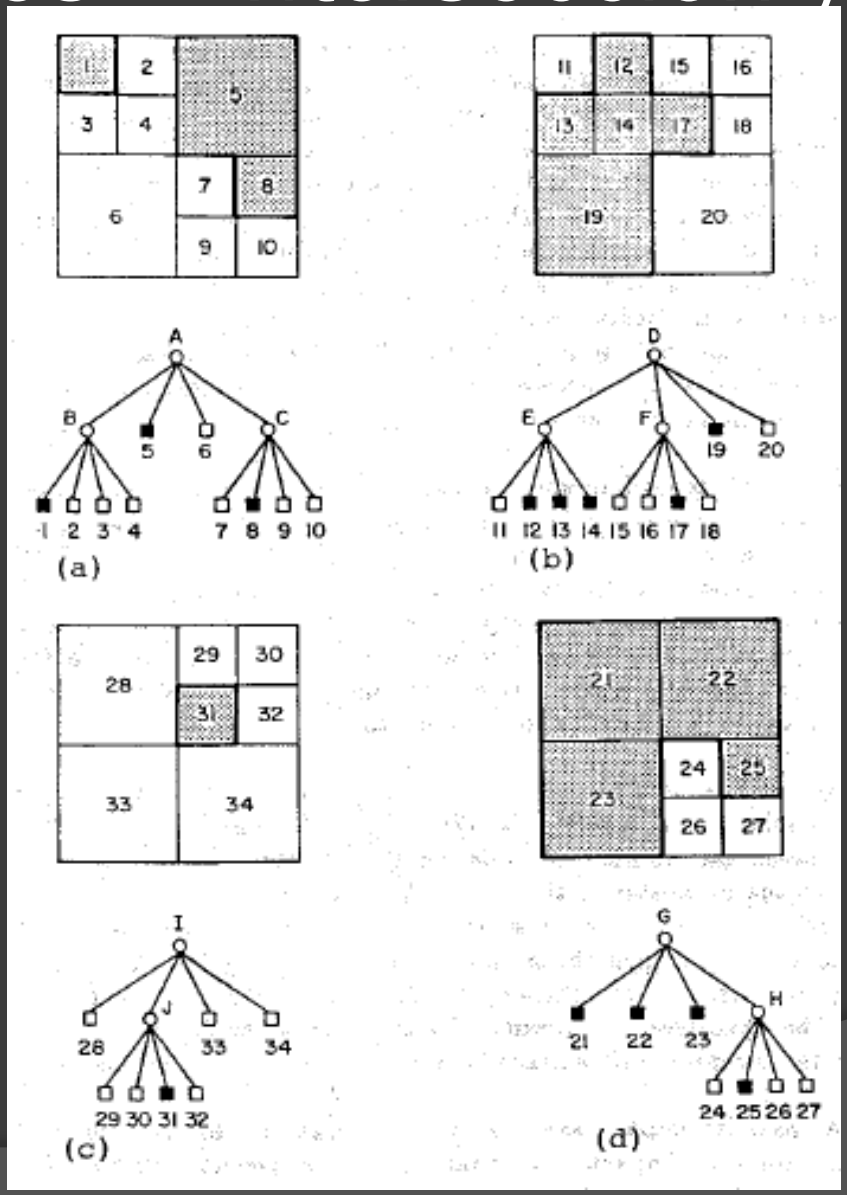
Quadtrees - Representación



Quadtrees - Intersección

- Si algún cuadrante es BLANCO, entonces el quadtree resultante es BLANCO
- Si ambos nodos son NEGRO, entonces la salida es NEGRO
- Si una entrada es NEGRO mientras las otras entradas son GRIS, entonces el subárbol del nodo GRIS es copiado en la salida del quadtree
- Si ambas entradas del quadtree son GRIS, entonces la salida del nodo es GRIS, y estas cuatro acciones son aplicadas recursivamente a cada par de hijos correspondiente..

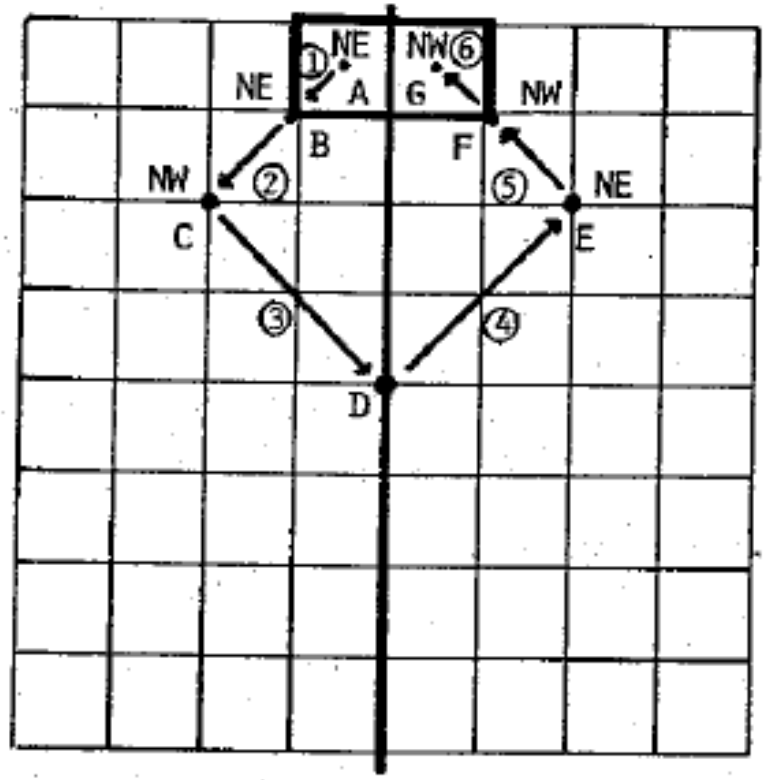
Quadtrees – Intersección y Unión



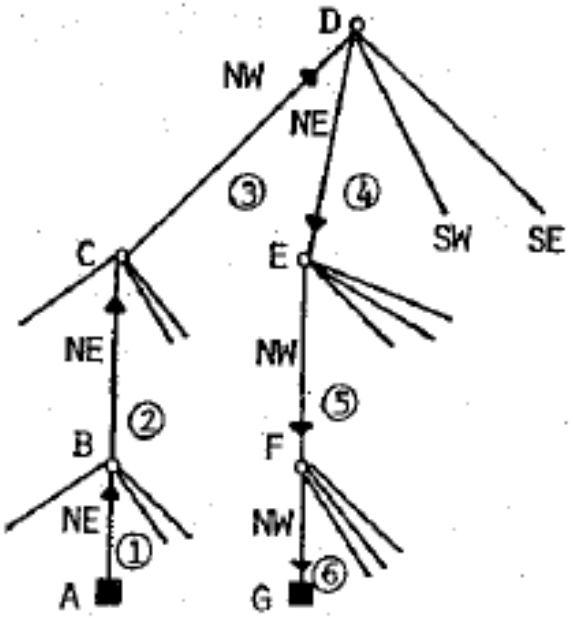
Quadtrees – Encontrar Vecinos

- Su vecino de igual tamaño en la dirección horizontal o vertical es determinado localizando un ancestro común. Luego, se recorre el camino mientras se hace una imagen espejo se mueve alrededor de un eje formado por el limite común entre los bloques asociados con los dos nodos.
- Primero, localizamos el ancestro más cercano al nodo dado que también sea adyacente a un ancestro del vecino buscado. Luego, hacemos uso de la función de búsqueda anterior para acceder los ancestros del vecino buscado en la dirección de adyacencia. . Finalmente, repasamos los pasos mientras hacemos movimientos directamente opuestos

Quadtrees – Encontrar Vecinos

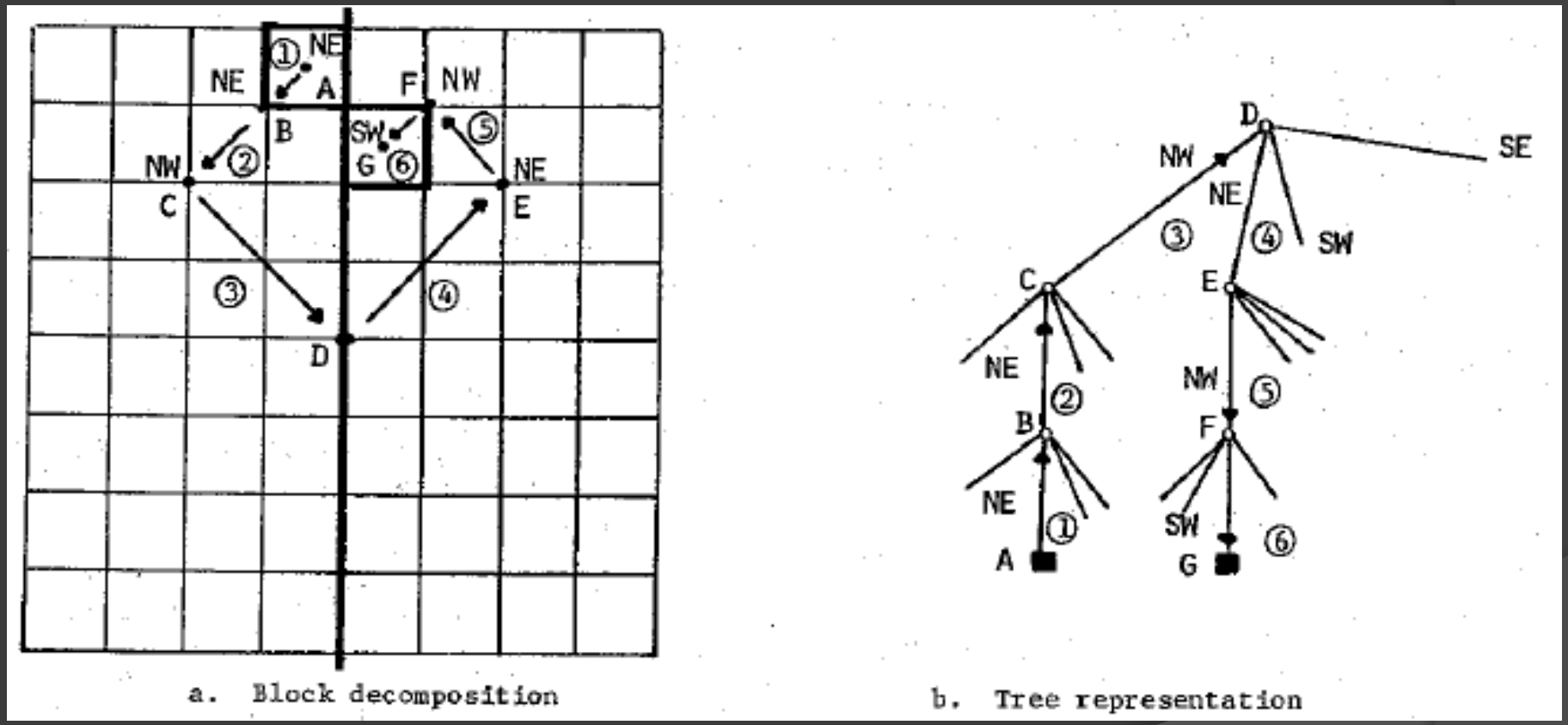


a. Block decomposition



b. Tree representation

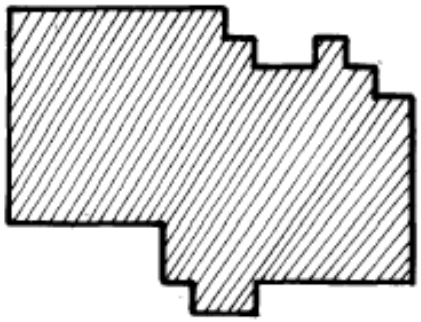
Quadtrees – Encontrar Vecinos



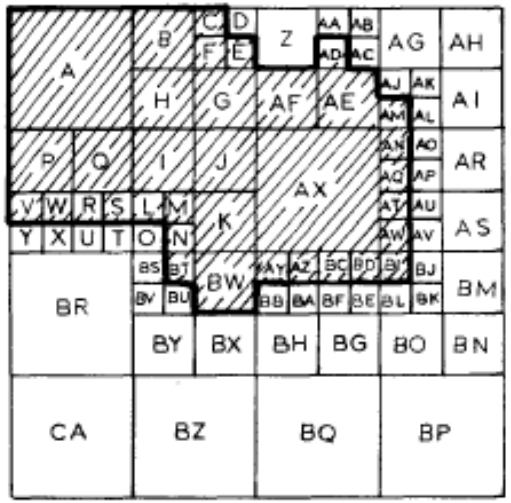
Quadtrees – Construir Quadtrees

- 1 Crear un nodo, llamado el nodo raíz, el cual representa la imagen en un todo
- 2 Ajustar la región de interés para que sea toda la imagen. Ajustar el nodo actual para que sea el nodo raíz
- 3.1 Si la región de interés es de un solo color, el nodo actual contenga ese color.
- 3.2.1 De otra manera, dividir la región de interés en cuatro sub-regiones dividiéndola en mitades verticales y horizontales.
- 3.2.2 Representar cada una de estas nuevas regiones por nodo hijo del nodo actual
- 3.2.3 Ajustar cada una de estas sub/regiones para que sea la región de interés, y su nodo para que sea el nodo actual, devolverse al paso 3.1 cada vez.

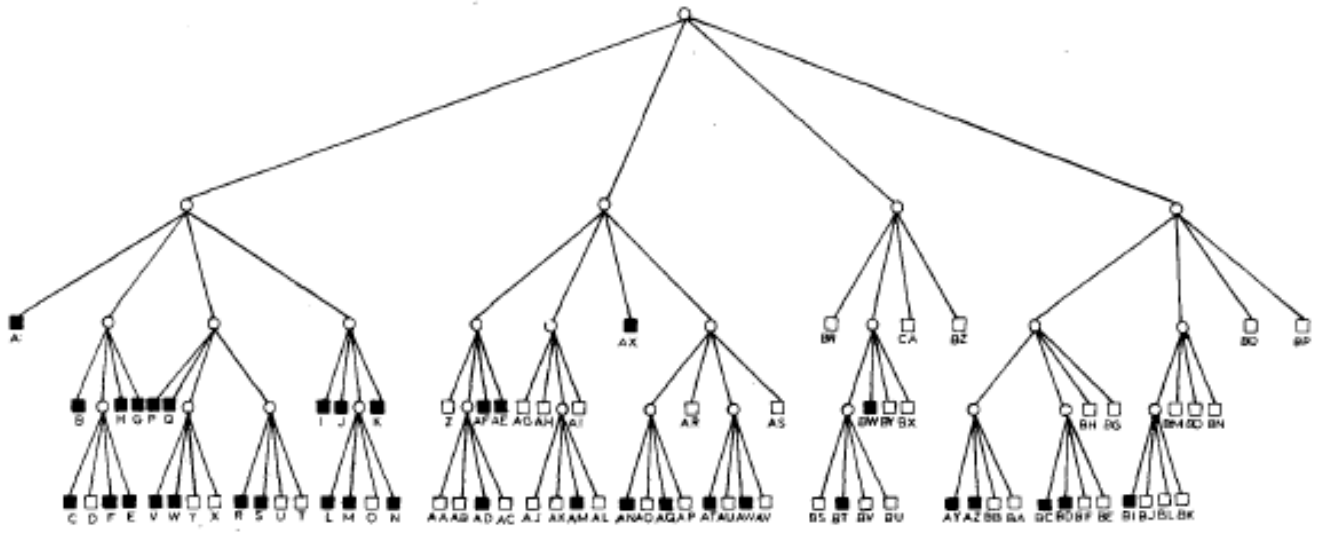
Quadtrees – Construir Quadtrees



(a) Sample region.



(b) Block decomposition of the region in (a).

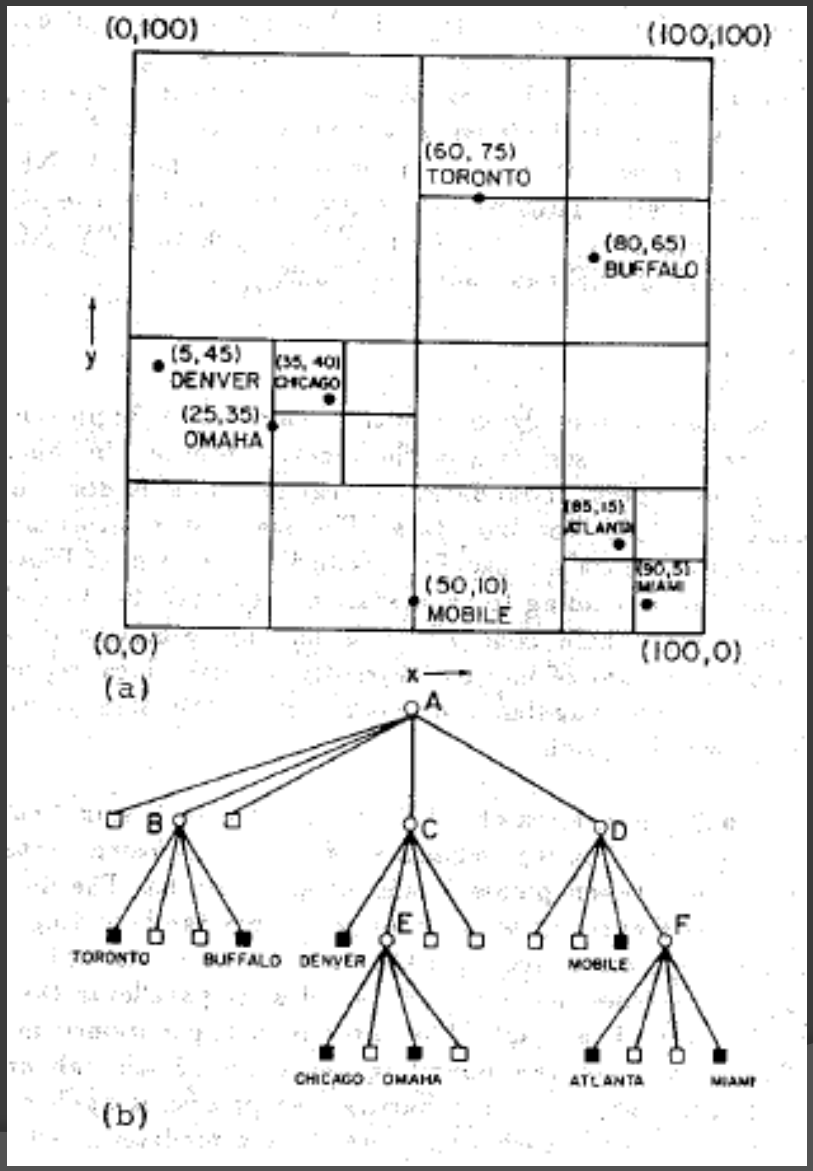


(c) Corresponding quadtree.

Quadtrees – Puntos de Datos

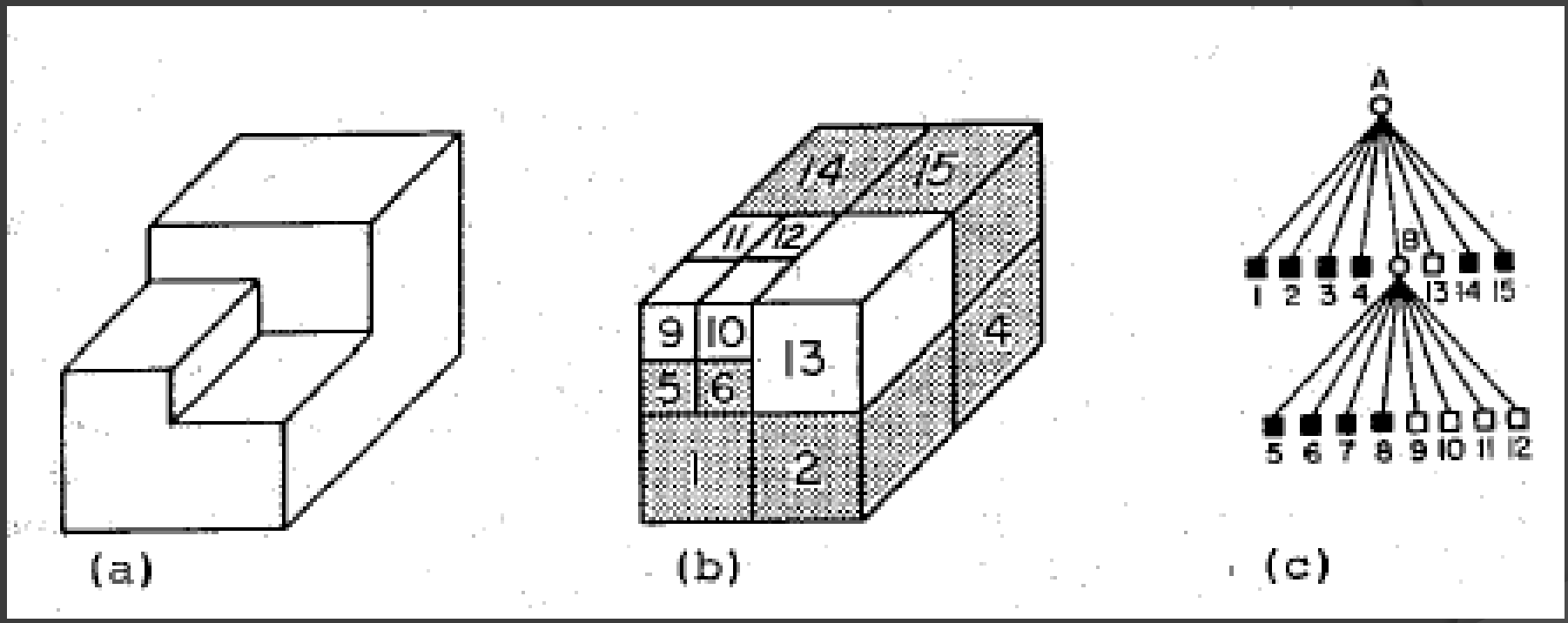
- Los PR quadtree son organizados de la misma manera que el quadtree de regiones. La diferencia está en que los nodos hojas son vacios (BLANCOS) o contienen un punto de información (NEGROS) y sus coordenadas. Un cuadrante contiene al menos un punto de información

Quadtrees – Puntos de Datos



Octrees

- Es un tipo de árbol, en el cual los nodos (excepto las hojas) cada uno tiene un máximo de 8 nodos hijos. La mayoría de los nodos representan volúmenes los cuales son subsecuentemente divididos en un máximo de 8 octantes representados por los nodos hijos


14
15
11
12
9
10
13
4
5
6
1
2
A
8
1 2 3 4 13 14 15
5 6 7 8 9 10 11 12
(a)
(b)
(c)

Edsger Dijkstra

“The effort of using machines to mimic the human mind has always struck me as rather silly. I would rather use them to mimic something better.”