Página Principal Mis cursos CCOMP6-1 - 63713 SEMANA 9: Del 18/10 al 23/10 | Examen Parcial Teórico

Examen Parcial Teórico (2021-II)



11/12/21 14:06 Pregunta 3 Correcta datos Puntúa 1.00 sobre 1.00 I. R-tree II. KD-tree III. R\*-tree V. Bkd-tree Seleccione una: b. VVFVF c. FFFFF d. VVVVV ● e. FFVVF Respuesta correcta Pregunta 4 Correcta Puntúa 1.00 sobre 1.00 la sobreposición. Seleccione una: Verdadero ■ Falso 

✓ Pregunta 5 Correcta Puntúa 1.00 sobre 1.00 dimensionalidad. Seleccione una: Verdadero Falso 

✓ Pregunta 6 Correcta Puntúa 1.00 sobre 1.00 Seleccione una:

Cuales de las siguientes Estructuras de datos utilizan la información geométrica (ordenamiento geométrico) de sus IV. R (Greene approach) - tree a. Ninguna de las anteriores La respuesta correcta es: FFVVF La variante del R-tree propuesta por Green (Green approach) propone introducir el cálculo del perímetro de las regiones para que estas sean cada vez mas rectangulares, y así se evite La respuesta correcta es 'Falso' La maldición de la dimensionalidad se refiere únicamente a la dificultad de extraer una muestra representativa de un conjunto de datos a medida que este va creciendo en La respuesta correcta es 'Falso' El pr-quadtree tiene como estrategia la partición del espacio, tomando como centro de dicha partición la posición del punto que se está insertando. Verdadero

La respuesta correcta es 'Falso'

■ Falso 

✓

11/12/21 14:06 Examen Parcial Teórico (2021-II): Revisión del intento Pregunta 7 Son estrategias para ordenar los octantes de un octree Correcta I. Nearest Neighbor Puntúa 1.00 II. Z-order sobre 1.00 III. Range Query IV. Bounding Box V. Todas las anteriores Seleccione una: a. VVFFF b. VVFFF c. FFFFF d. FVFFF e. VVVVV Respuesta correcta La respuesta correcta es: FVFFF Pregunta 8 El árbol R+ (R+ tree) toma en consideración el área, perímetro y Correcta la sobreposición de las nuevas regiones a ser creadas durante su función de split (Node splitting) Puntúa 1.00 sobre 1.00 Seleccione una: Verdadero Falso La respuesta correcta es 'Falso'

Pregunta **9**Finalizado
Puntúa 0.50
sobre 6.00

En un implementación de R\*-Tree (RStar), ya se tiene finalizadas las siguientes funciones (listas para ser llamadas)

- RStar::ChooseSubTree(Rect input): Que devuelve un puntero a un RStarNode donde debería ser insertado el dato en cuestión.
- RStar::Delete(Rect a): Que borra el rectangulo a, si es que existe
- RStar::Find(Rect a): Que encuentra y devuelve un puntero al RStarNode donde se encuentra dicho rectangulo, si es que está presente en el árbol.

Además se usa las siguientes clases:

- class *Rect*, que contine los valores x\_min, x\_max, y\_min,
   y\_max (*float*) que describen el rectangulo, que será nuestro dato a ser indizado en nuestra estructura de datos.
- class *RStarNode,* que contiene un *std::vector<Rect> data* almacenando los rectangulos contenidos en dicho nodo.

Con la información anteriormente detallada:

- Implemente en C++ la función
   RStar::OverflowTreatment(RStarNode \*node), incluyendo sus dos funciones internas ReInsert y Split.
- 2. Se recomienda adjuntar un archivo .cpp, sin embargo puede copiar todo el código en el cuadro de texto, manteniendo lo mas posible la indentación.
- 3. **IMPORTANTE**: Dejar *comentarios* en el código para entender que acción está intentando realizar.

Archivo adjunto























## <u>pregunta9andres.cpp</u>

## Comentario:

- El alumno intentó implementar el primer paso de la función Reinsert, sin embargo olvidó previamente calcular el centro del parámetro "node", luego recién se calculan las distancias a este.
- No se ha implementado la función Split()
- La sentencia if(root->leaves ==0){ // La primera llamada de overflowtratment . no es correcta pues se debe confirmar que NO se ha llamado anteriormente a la función ReInsert en el NIVEL del arbol donde está "node", no en la raiz.



Pregunta 10

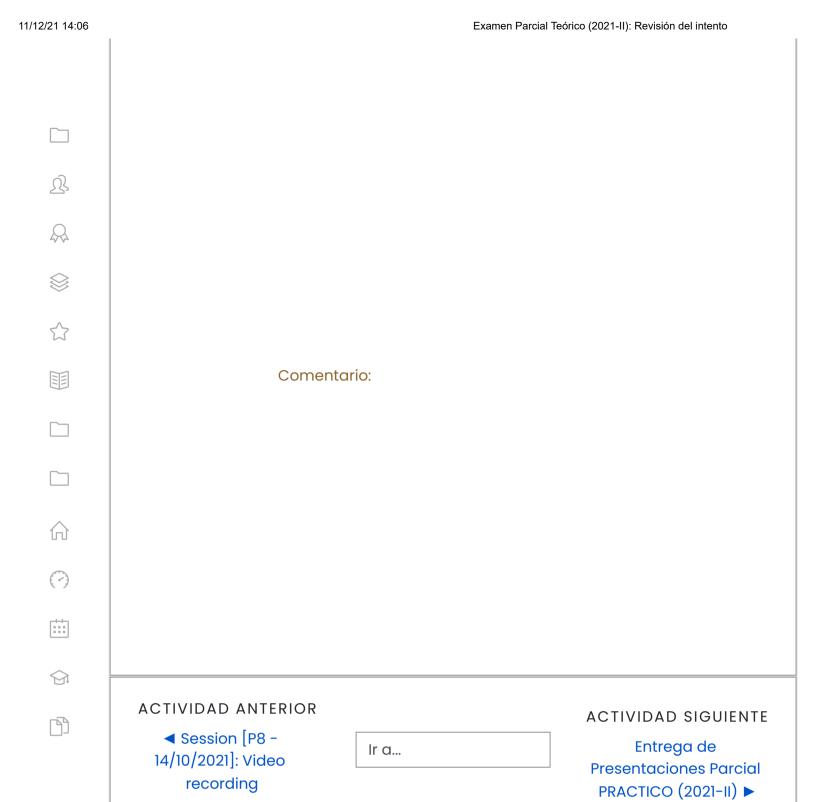
Finalizado

Puntúa 0.00 sobre 6.00

11/12/21 14:06 

Usando la estructura de datos KD-tree, implemente una función para dar soporte a la búsqueda ortogonal en rango como se observa en la **Figura 1 (izquierda)**. Usar la sintaxis desarrollada en laboratorio: std::vector<ElemType> ortogonal\_range\_query(const std::vector<Pair<double, double>> &ranges) const; Retorna un vector de los valores contenidos en el hipercubo, el cual está formado por un vector ranges de tamaño N cuyos elementos son cada uno un par de double que delimitan al hipercubo en cada una de las dimensiones, Ejemplo vector con N=2 [<2, 8>; <1, 4>] forma el siguiente cuadrado Figura 2 (derecha).

no realizada



## Mantente en contacto

Universidad Católica San Pablo

- https://ucsp.edu.pe

