

Teoría de árboles a través del uso del software *Mathematica*

Mag. Enrique Vílchez Quesada
Universidad Nacional de Costa Rica
enrique.vilchez.quesada@una.cr

Resumen: con el presente trabajo se comparte uno de los resultados de investigación de un proyecto adscrito a la Escuela de Informática de la Universidad Nacional de Costa Rica, titulado: “Facebook como herramienta de enseñanza y aprendizaje para el curso *EIF-203 Estructuras Discretas para Informática* a través del uso de cuadernos interactivos”. Parte de las tareas vinculadas consistieron en desarrollar una serie de experiencias de aprendizaje para abordar la teoría de árboles utilizando como apoyo el conocido software comercial *Mathematica*. El presente documento integra una serie de ejemplos que pueden servir como modelo experimental hacia la búsqueda de una metodología asistida por computadora, en un área de conocimiento con una incidencia pedagógica usualmente basada en la clase magistral.

Objetivo general:

Desarrollar la teoría de árboles utilizando como recurso pedagógico el software *Mathematica*.

Objetivos específicos:

1. Mostrar los principales comandos que integra el software *Mathematica* en el campo de la teoría de árboles.
2. Explicar distintas funciones creadas por el autor de esta propuesta, que complementan las funciones por default de *Mathematica* en teoría de árboles.
3. Mostrar la programación creada por el autor en el entorno de *Mathematica*, de distintos algoritmos clásicos de la teoría de árboles.
4. Compartir ejemplos asistidos por computadora hacia la búsqueda de una metodología innovadora en este campo.

Recursos y materiales

Se entregará a los participantes un CD con el contenido del taller, los ejemplos asistidos por computadora, las funciones creadas por el autor del presente trabajo y diversos algoritmos clásicos programados en *Mathematica*.

Introducción

La teoría de árboles en cursos relacionados con matemática discreta usualmente es abordada con un enfoque de resolución de problemas en papel. Esta metodología tradicional ocasiona con frecuencia el desarrollo de ejercicios triviales y sin ninguna aplicación práctica, principalmente en carreras vinculadas con ingeniería. Torres resalta este vacío metodológico y su complejidad de cambio al señalar: “*una de las principales dificultades ... tiene que ver con la producción de situaciones problemáticas relacionadas con el mundo real*” (2006, p. 754) y a este respecto, la enseñanza de la matemática asistida por computadora puede brindar algunas respuestas.

El presente trabajo busca ofrecer una nueva alternativa metodológica utilizando como principal recurso de investigación y resolución de problemas el software comercial *Mathematica*. *Mathematica* es una aplicación que incorpora un paquete denominado “combinatorica”, éste contempla cerca de 450 funciones en el campo de las estructuras discretas (Pemmaraju y Skiena, 2009), algunas de ellas específicamente para resolver problemas típicos de la teoría de árboles. Combinatorica no ofrece aún así, todas las respuestas a ejercicios usuales dentro de esta teoría, razón por la cual se han desarrollado algunas funciones personales para tratar algunos temas particulares.

Otra ventaja que ofrece el entorno de programación de *Mathematica* es la relativa facilidad con la que permite la implementación de algunos algoritmos tradicionales de la teoría de árboles, entre los que se destacan: los códigos de *Huffman*, árboles binarios de búsqueda, el algoritmo de *Prim* y el algoritmo de *Kruskal*.

Este trabajo contribuye con una aproximación pedagógica para docentes interesados en innovar a través del uso de tecnología como un complemento de la clase magistral.

Contenidos

Se desarrollarán los siguientes contenidos con apoyo de software:

- Introducción
- Árboles binarios de búsqueda
- Recorridos en un n -árbol
- Árboles generadores
- Árboles de expansión mínima

Metodología

Se iniciará el taller compartiendo algunos comandos básicos del software *Mathematica* para tratar la teoría de árboles. Posteriormente, se irá profundizando mediante ejemplos más abiertos asistidos por computadora, recurriendo al lenguaje de programación de la aplicación utilizada. A través de un archivo facilitado a los participantes, se irán desarrollando cada uno de los ejemplos expuestos en la guía de trabajo presentada más adelante.

Descripción de actividades

Las actividades a implementar se resumen en la siguiente guía de trabajo. En general el avance de los participantes se regulará mediante las intervenciones oportunas del expositor apoyando y dando seguimiento al trabajo realizado.

Guía de trabajo

1. Se iniciará con una explicación del uso de los comandos básicos: `TreeGraph`, `TreePlot`, `RandomTree`, `KaryTree`, `CompleteKaryTree`, `CompleteBinaryTree`, `TreeGraphQ`, `TreeForm`, `DepthFirstScan`, `BreadthDepthFirstScan`, `BreadthFirstTraversal`, `DepthFirstTraversal` y `MinimumSpanningTree`.
2. Se resolverán algunos ejercicios asistidos por computadora, entre ellos:
 - a. A través de la instrucción `RandomGraph`, desarrolle una función que forme árboles pseudoaleatorios con n vértices.
 - b. Verifique la correctitud de la función del ejemplo anterior utilizando el comando `TreeGraphQ`.
 - c. Diseñe una función en *Mathematica* que devuelva la lista de todos los nodos terminales de un árbol recibido como parámetro.
 - d. Elabore una función que determine la altura de un árbol.
 - e. Halle con ayuda de *Mathematica*, el número de vértices y lados en un k -árbol completo de n niveles.
 - f. Verifique la veracidad del siguiente enunciado: sea T un árbol binario completo con i vértices internos entonces T contiene $i \lfloor \frac{1}{2} \rfloor$ hojas y $2i \lfloor \frac{1}{2} \rfloor$ nodos en total.
 - g. Implemente en *Mathematica* el algoritmo de códigos de *Huffman* optimizado.
 - h. Elabore tres funciones booleanas distintas en *Mathematica*, que reciban como

parámetro un grafo T y determinen si T es un árbol.

- i. Construya una función booleana con *Mathematica* que concluya si un grafo recibido como parámetro es un árbol binario.
- j. Usando la instrucción `RandomGraph`, genere una función que construya árboles binarios pseudoaleatorios con n vértices.
- k. Genere una función que construya árboles binarios de búsqueda.
- l. Construya con software la notación polaca y polaca inversa de

$\mathbb{M} \Rightarrow b \cup c \Rightarrow d \cup e \cup \mathbb{M} \Rightarrow b \cup c \Rightarrow j \cup$

- m. Construya en *Mathematica* una función que reciba como parámetro un grafo G y devuelva un árbol generador por cada vértice, retornando además, el número de árboles de expansión que posee.
- n. Elabore en *Mathematica* un programa que resuelva la búsqueda a lo largo de un dato w sobre un k -árbol.
- o. Implemente el algoritmo de *Prim* en *Mathematica*.

Con la finalidad de brindar una cobertura apropiada a las actividades anteriores, se darán resueltos varios de los ejercicios descritos con anterioridad, con la opción de modificar algunos parámetros. Por ejemplo, un programa que resuelve el inciso k. es:

```
W = {};  
Ordena[Datos_List] :=  
Module[{CNI = {}, CND = {}, L = DeleteDuplicates[Datos]},  
  If[Length[L] ≥ 2, For[i = 2, i ≤ Length[L],  
    If[L[[1]] > L[[i]], CNI = Append[CNI, L[[i]]],  
    CND = Append[CND, L[[i]]]; i++];  
  If[CNI ≠ {}, W = Append[W, L[[1]] → CNI[[1]]];  
  If[CND ≠ {}, W = Append[W, L[[1]] → CND[[1]]]; Ordena[CNI];  
  Ordena[CND]]]  
Datos = ;  
Ordena[Datos]  
TreePlot[W, Automatic, Datos[[1]], VertexLabeling → True,  
  ImagePadding → 10]
```

Este código será compartido directamente a través de un notebook incluido en el CD a distribuir a los participantes del taller. Como se observa, en él la variable “Datos” no contiene ningún vector. La idea en este sentido es que el usuario prueba la función “Ordena” con el conjunto de datos que desee. Esta función construye un árbol binario de búsqueda automáticamente con los valores incluidos en “Datos”.

De manera similar se espera resolver las otras actividades descritas, con la intención de facilitar el adecuado desarrollo del taller y garantizar su completitud al finalizar el tiempo designado.

Como un aporte adicional, se entregará a los participantes un cuaderno interactivo diseñado por el autor de esta propuesta, que explica en audio y en formato PDF cada uno de los programas a desarrollar. El cuaderno constituye un respaldo multimedia del contenido del taller. Éste presenta el siguiente formato:

Ejemplo A través de la restricción `RandomGraph`, desarrolle una función que forme árboles pseudoaleatorios con (n) vértices.
 Ya se había creado la función `ArbolSAConexo`
`ArbolSA[n_] := Module[{t}, t = RandomGraph[n, n - 1];`
`VertexLabels -> "Name", ImagePadding -> 10];`
`Write[ConnectedGraphQ[t] == False,`
`t = RandomGraph[n, n - 1], VertexLabels -> "Name",`
`ImagePadding -> 10]; Return[t];]`

$n = 20$

```

graph TD
    11 --- 10
    11 --- 16
    11 --- 19
    11 --- 12
    10 --- 18
    10 --- 5
    18 --- 1
    18 --- 20
    18 --- 7
    16 --- 3
    16 --- 14
    3 --- 15
    3 --- 17
    15 --- 9
    15 --- 2
  
```

Otra Sentencia interesante es: `TreeGraphQ`

Ejemplo Verifique la conectividad de la función `ArbolSA` utilizando el comando `TreeGraphQ`.
 En *Mathematica*:
`n = 0;`
`For[{i, j}, i < 1000, {j}, If[TreeGraphQ[ArbolSA[i, j]] == False, n = 1,`
`Break[{i, j}]; i++;]`
`If[n == 0, Print[n == n todos los casos se formó un árbol];]`
`Print[n == n al menos un caso no se formó un árbol];]`

Como se visualiza en la figura, el multimedia consiste literalmente en un cuaderno donde el usuario al colocar el mouse en cualquier lugar con un puntero circular, puede adelantar o bien retroceder de manera dinámica explicaciones en audio de los ejemplos resueltos con *Mathematica*.

Referencias bibliográficas

- Ávila, J. (2005). *Estructuras de matemática discreta para computación*. Costa Rica: UNA.
- Torres, C. (2006). *Enseñanza de la trigonometría asistida con calculadoras graficadoras*. En Ricardo Cantoral Uriza, Olda Covián Chávez et al. (Comps.), *Investigaciones sobre enseñanza y aprendizaje de las matemáticas* (pp. 753-778). México: Ediciones Díaz de Santos, S.A.
- Johnsonbaugh, R. (2005). *Matemáticas discretas*. México: Pearson Prentice Hall.
- Pemmaraju, S. & Skiena, S. (2009). *Computational discrete mathematics: combinatorics and graph theory with Mathematica*. USA: Cambridge University Press.
- Vílchez, E. (2013). *Estructuras Discretas con Mathematica*. Costa Rica: Escuela de Informática de la Universidad Nacional de Costa Rica.