

Resultados del modo de ejemplo

I. Descripción del problema

Se tienen 6 llaves diferentes (A, B, C, D, E y F), cada una de las cuales tiene una probabilidad diferente de ser buscada. El objetivo es construir un árbol de búsqueda binario a partir de estas llaves, de forma que el tiempo de búsqueda promedio sea el menor posible. Para esto se empleó un algoritmo de programación dinámica y un algoritmo greedy.

II. Datos del problema

Los datos utilizados se resumen en la siguiente tabla:

Llave	A	B	C	D	E	F
Probabilidad	0.15	0.14	0.21	0.10	0.11	0.29

III. Programación Dinámica

A continuación se muestran las tablas generadas por el algoritmo de programación dinámica para encontrar el árbol óptimo. Además, se muestra el árbol resultante.

Tiempo de ejecución: 0.000003 segundos.

$i \backslash j$	0	1	2	3	4	5	6
1	0.00	0.15	0.43	0.86	1.13	1.45	2.24
2		0.00	0.14	0.49	0.69	1.01	1.80
3			0.00	0.21	0.41	0.73	1.41
4				0.00	0.10	0.31	0.81
5					0.00	0.11	0.51
6						0.00	0.29
7							0.00

Table 1: Matriz A generada por el algoritmo de programación dinámica

$i \backslash j$	1	2	3	4	5	6
1	1	1	2	3	3	3
2		2	3	3	3	3
3			3	3	3	5
4				4	5	6
5					5	6
6						6

Table 2: Matriz R generada por el algoritmo de programación dinámica

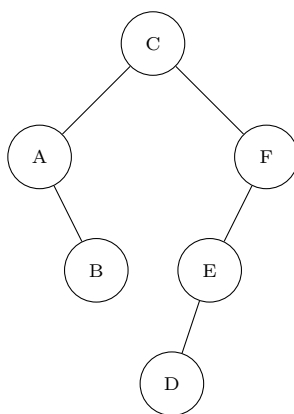


Figure 1: Árbol generado por el algoritmo de programación dinámica

IV. Algoritmo Greedy

A continuación se muestran la tabla R generada por el algoritmo greedy. Además, se muestra el árbol resultante.

Tiempo de ejecución: 0.000002 segundos.

$i \backslash j$	1	2	3	4	5	6
1	1	1	3	3	3	6
2		2	3	3	3	6
3			3	3	3	6
4				4	5	6
5					5	6
6						6

Table 3: Matriz R generada por el algoritmo greedy

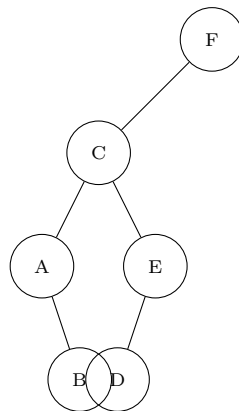


Figure 2: Árbol generado por el algoritmo greedy