Instituto Tecnológico de Costa Rica

PROYECTO DE INVESTIGACIÓN DE OPERACIONES

PROYECTO 2

Árboles Binarios

Estudiantes

Jason Barrantes Arce
2015048456

Steven Bonilla Zúñiga
2015056296

Profesor

Francisco Torres Rojas

Modo Ejemplo:

Se resolverá un problema general por medio de diversos algoritmos que nos permitan encontrarle una solución. El problema es sobre árboles de búsqueda binaria.
Hay que ordenar una serie de valores o llaves en forma de árbol binario, de manera que el nivel de búsqueda promedio sea el óptimo. Tenemos una serie de llaves que quieren acomodarse en una estructura de árbol, pero esas llaves deben tener un respectivo peso, que determinará su probabilidad y un cáracter único que será asignado de manera ascendente.

Restricciones:

- **Estructura:** Se formará un árbol binario óptimo.
- **Llaves:** Se generarán 6 llaves de forma ascendente.
- **Peso:** Varía entre $1 \leq Ci \leq 1000$.
- Los caracteres ASCII varían.

Los dos algoritmos que vamos a implementar son:

- **Algoritmo de Búsqueda Dinámica:** Algoritmo para el ABB óptimo.
- **Algoritmo Greedy Básico:** Cada vez se escoge la llave de máxima probabilidad para que sea la raíz del árbol.

En el caso de programación dinámica ya que nuestro objetivo es minimizar el costo promedio de la búsqueda, usamos la fórmula:

$$\text{MIN(Z)} = \sum_{i=1}^{n} c_i p_i$$

Que está sujeto a:

$$\sum p_i \equiv 1$$

Con cada $c_i = 1, 2, \ldots n$

Se muestra a continuación la tabla de objetos con su respectivo costo (peso) y probabilidad que fueron asignados aleatoriamente cumpliendo con las restricciones:

Letra	**Peso**	**Probabilidad**
Object A	891	0.28
Object B	562	0.18
Object C	536	0.17
Object D	67	0.02
Object E	792	0.25
Object F	360	0.11

1. Algoritmo AAB Óptimo

Es un problema de solución de búsqueda óptima: En nuestro caso, queremos minimizar el costo de la búsqueda promedio. Para solucionar el problema vamos a hacer uso de una tabla (n+1) x (n+1), donde n es la cantidad de objetos o llaves disponibles. Como se menciona en las restricciones del problema $n = 6$ por lo que tendremos dos tablas (7x7). La tabla A donde estará el costo promedio y la tabla R donde estará los índices de búsqueda más rápidos.

Fórmula Matemática

$$\text{MIN(Z)} = \sum_{i=1}^{6} c_i p_i$$

Sujeto a:

$$\sum p_i \leq 1$$

En otras palabras tendremos

$$(p_1 \approx 0{,}2777) + (p_2 \approx 0{,}1752) + (p_3 \approx 0{,}1671) + (p_4 \approx 0{,}0209) + (p_5 \approx 0{,}2469) + (p_6 \approx 0{,}1122) = 1$$

Ahora proseguimos realizando la tabla dinámica.

Tabla A:

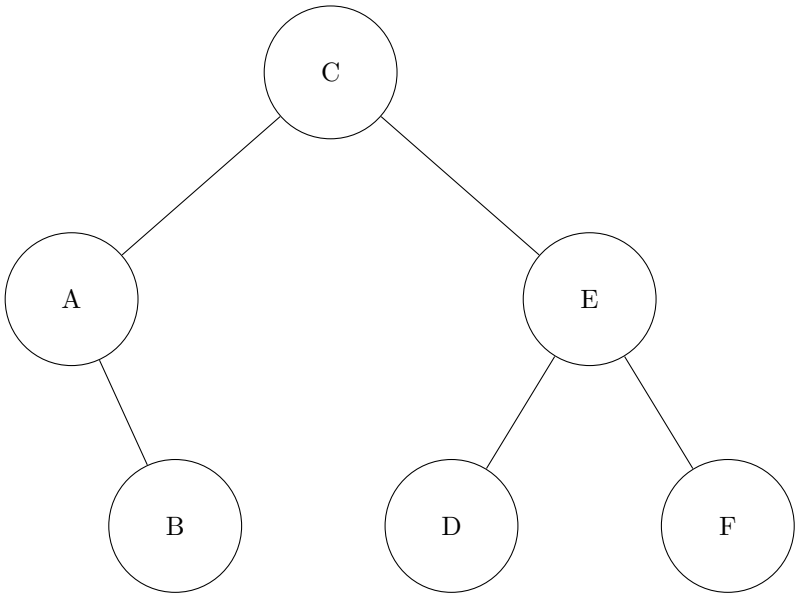
X	**0**	**1**	**2**	**3**	**4**	**5**	**6**
1	0.00	0.28	0.63	1.06	1.13	1.80	2.14
2	–	0.00	0.18	0.51	0.56	1.07	1.39
3	–	–	0.00	0.17	0.21	0.64	0.87
4	–	–	–	0.00	0.02	0.29	0.51
5	–	–	–	–	0.00	0.25	0.47
6	–	–	–	–	–	0.00	0.11
7	–	–	–	–	–	–	0.00

Tabla R:

X	**0**	**1**	**2**	**3**	**4**	**5**	**6**
1	0	1	1	2	2	3	3
2	-	0	2	2	3	3	5
3	-	-	0	3	3	5	5
4	-	-	-	0	4	5	5
5	-	-	-	-	0	5	5
6	-	-	-	-	-	0	6
7	-	-	-	-	-	-	0

El algoritmo tarda aproximadamente: 0.003000 segundos en ejecutarse.

El árbol de búsqueda binario queda representado de la siguiente forma:



2. Algoritmo Greedy Básico

Es un algoritmo que soluciona problemas que a primera vista parece ser óptimo. Es característico porque es muy sencillo de entender y explicar. Se escoge la llave de máxima probabilidad para que sea la raíz del árbol, las demas se separan en dos grupos: las menores que la raíz y las mayores. Este procedimiento se repite recursivamente en cada grupo.

$$Llave_i = (Peso, Probabilidad), i = 0...n$$

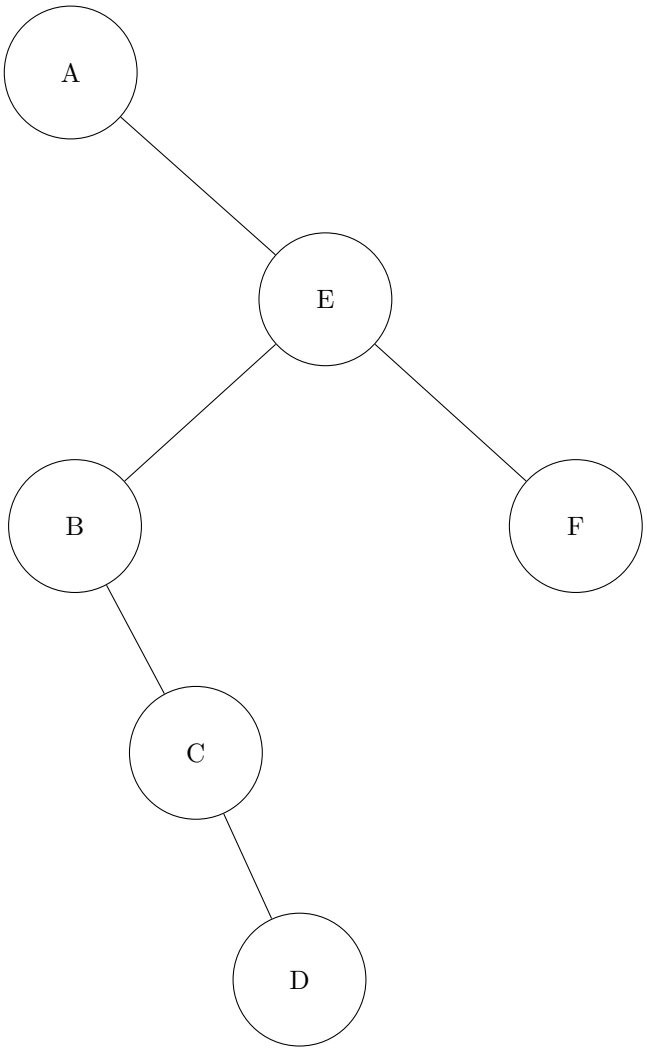
$K_A = (891, 0{,}28), K_B = (562, 0{,}18), K_C = (536, 0{,}17), K_D = (67, 0{,}02), K_E = (792, 0{,}25), K_F = (360, 0{,}11)$

Ahora proseguimos realizando la tabla greedy básico. **Tabla R:**

X	**0**	**1**	**2**	**3**	**4**	**5**	**6**
1	0	1	-	-	-	-	1
2	-	0	2	-	2	-	5
3	-	-	0	3	3	-	-
4	-	-	-	0	4	-	-
5	-	-	-	-	0	5	-
6	-	-	-	-	-	0	6
7	-	-	-	-	-	-	0

El algoritmo tarda aproximadamente: 0.047000 segundos en ejecutarse.

El árbol de búsqueda binario queda representado de la siguiente forma:


A
E
B
F
C
D