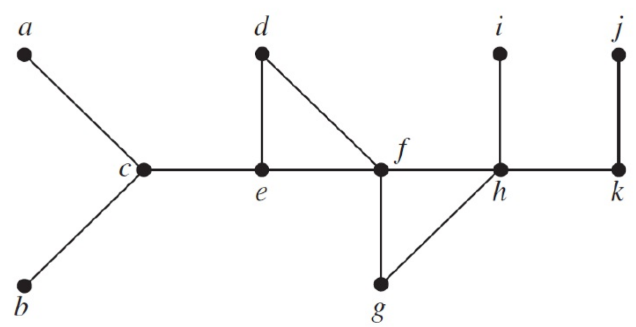
Búsqueda a lo ancho (BFS Breadth-First Search)

Input: Un grafo conexo G con conjunto de vértices $\{v_1, \ldots, v_n\}$.
Output: Un árbol de expansión T.
Iteración:

1. $T = \{v_1\}$
2. $L = [v_1]$
3. Mientras $L \neq \emptyset$
   a. Elimine el primer vértice v de L
   b. Para cada vecino w de v
      1. Si $w \notin L$ y $w \notin V(T)$
         a. Concatene w al final de la lista L
         b. $V(T) = V(T) \cup \{w\}$ y $E(T) = E(T) \cup \{vw\}$

G:

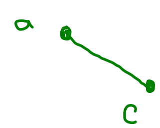


$T = \{a\}$

Visit(a)

$V(T) = \{a\} \cup \{c\}$

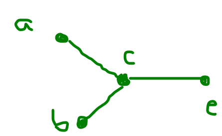
$E(T) = \phi \cup \{ac\}$



Visit(c)

$V(T) = \{a, c\} \cup \{b, e\}$

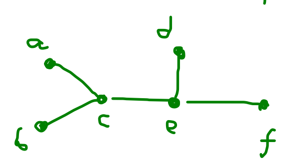
$E(T) = \{ac, cb, ce\}$



Visit(e)

$V(T) = \{a, c, b, e\} \cup \{d, f\}$

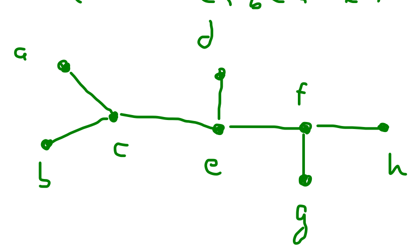
$E(T) = \{ac, cb, ce, ed, ef\}$



Visit(f)

$V(T) = \{a, c, b, e, d, f\} \cup \{g, h\}$

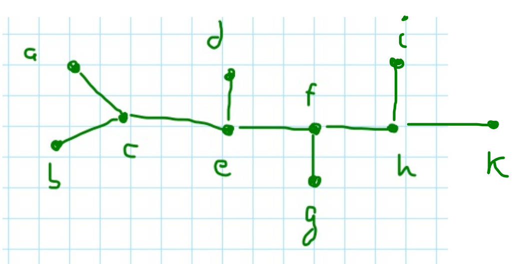
$E(T) = \{ac, bc, ce, ed, ef\} \cup \{fg, fh\}$



Visit(h)

$V(T) = \{a, c, b, e, d, f, g, h\} \cup \{i, k\}$

$E(T) = \{ac, bc, ce, ed, ef, fg, fh\} \cup \{hi, hk\}$

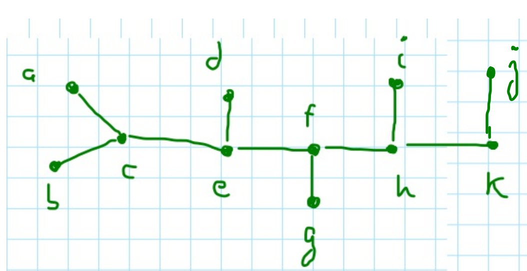


Visit(k)

$V(T) = \{a, c, b, e, d, f, g, h, i, k\} \cup \{j\}$

$E(T) = \{ac, bc, ce, ed, ef, fg, fh, hi, hk, kj\}$

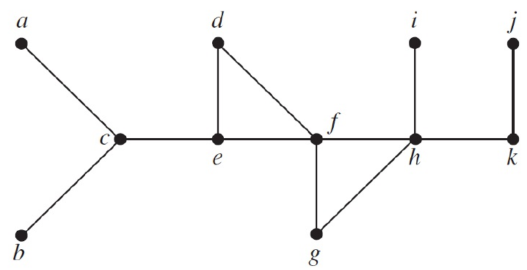
T



Búsqueda en profundidad (Backtracking)

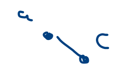
Input: Un grafo conexo G con conjunto de vértices $\{v_1, \dots, v_n\}$.
Output: Un árbol de expansión T.
Iteración:

1. $T = \{v_1\}$
2. *Visita*(v)
   a. Para cada vértice w adyacente a v, $w \notin V(T)$
      1. $V(T) = V(T) \cup \{w\}$ y $E(T) = E(T) \cup \{vw\}$
      2. *Visita*(w)

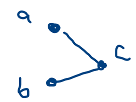


1. $T = \{a\}$

2. Visit(a)
   a. c
      1. $V(T) = \{a\} \cup \{c\}$
      2. $E(T) = \emptyset \cup \{ac\}$

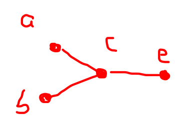


2. Visit(c)
   a. b
      1. $V(T) = \{a, c\} \cup \{b\}$
      2. $E(T) = \{ac\} \cup \{cb\}$

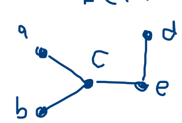


2. Visit(b) X

2. Visit(c)
   a. e
      1. $V(T) = \{a, c, b\} \cup \{e\}$
      2. $E(T) = \{ac, cb, ce\}$

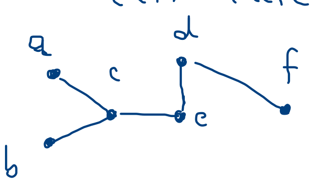


2. Visit(e)
   a. d
      1. $V(T) = \{a, c, b, e\} \cup \{d\}$
      2. $E(T) = \{ac, cb, ce\} \cup \{ed\}$



2. Visit(d)
   a. f
      1. $V(T) = \{a, c, b, e, d\} \cup \{f\}$

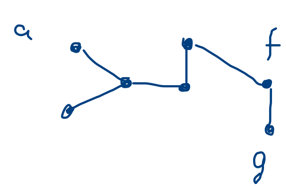
      $E(T) = \{ac, cb, ce, ed, df\}$



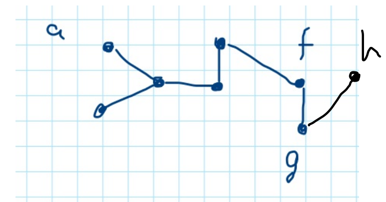
2. Visit(f)
   a. g

   $V(T) = \{a, c, b, e, d, f\} \cup \{g\}$

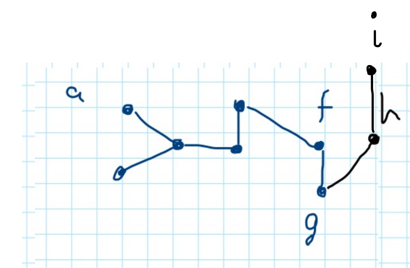
   $E(T) = \{ac, cb, ce, ed, df\} \cup \{fg\}$



2. Visit(g)
   a. h
      1. $V(T) = \{a, b, c, d, e, f, g\} \cup \{h\}$
      2. $E(T) = E(T) \cup \{gh\}$

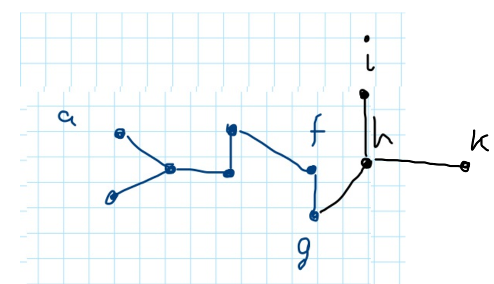


2. Visit(h)
   a. i
      1. $V(T) = V(T) \cup \{i\}$
      2. $E(T) = E(T) \cup \{hi\}$

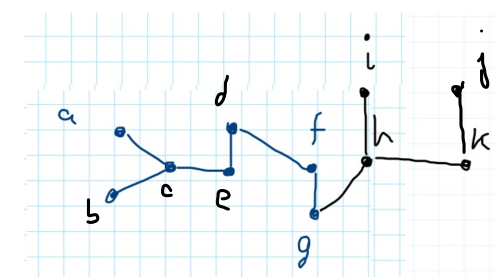


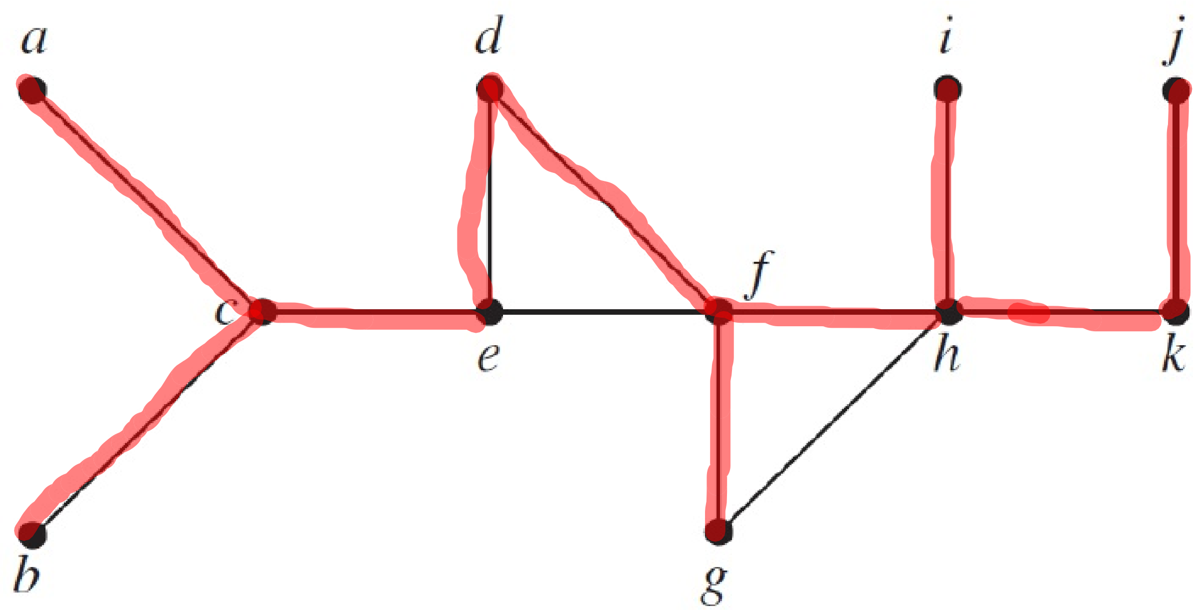
2. Visit(i) X

2. Visit(h)
   a. k
      1. $V(T) = V(T) \cup \{k\}$
      2. $E(T) = E(T) \cup \{hk\}$



2. Visit(k)
   a. j
      1. $V(T) = V(T) \cup \{j\}$
      2. $E(T) = E(T) \cup \{kj\}$





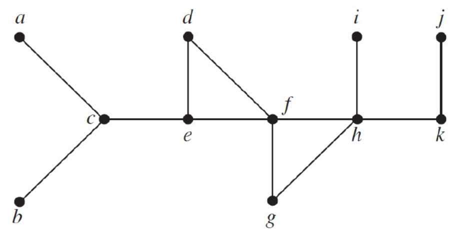
It	Vértice	Arista
0	h	—
1	f	(h, f)
2	d	(f, d)
3	e	(d, e)
4	c	(e, c)
5	a	(c, a)
6	b	(c, b)

It	V	Arista
7	g	(f, g)
8	i	(h, i)
9	k	(h, k)
10	j	(k, j)

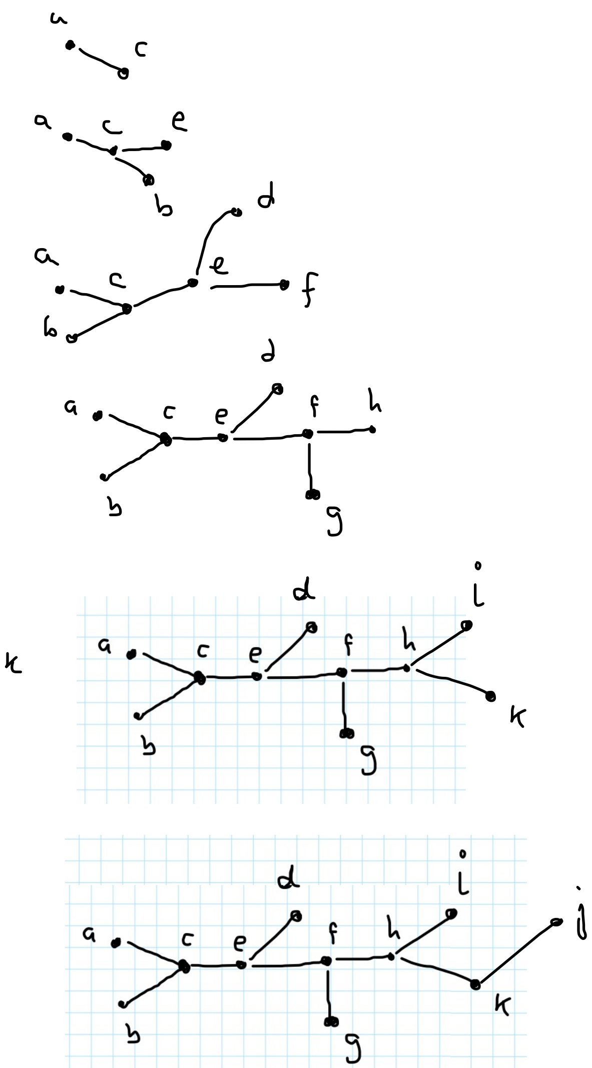
Búsqueda a lo ancho (BFS Breadth-First Search)

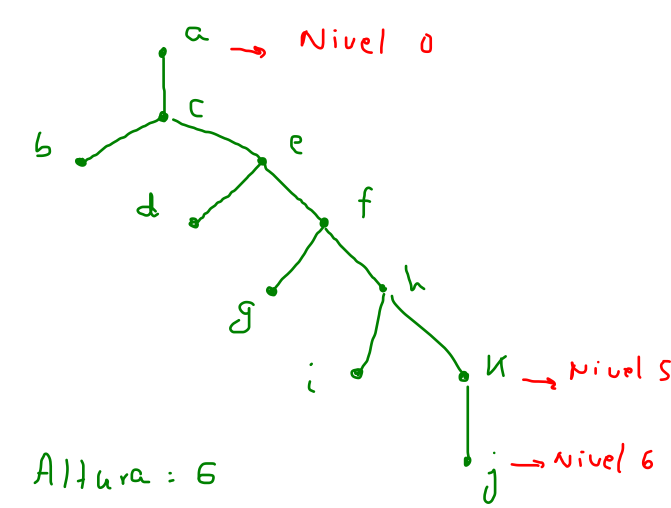
Input: Un grafo conexo G con conjunto de vértices $\{v_1, \dots, v_n\}$.
Output: Un árbol de expansión T.
Iteración:

1. $T = \{v_1\}$
2. $L = [v_1]$
3. Mientras $L \neq \emptyset$
    a. Elimine el primer vértice v de L
    b. Para cada vecino w de v
        1. Si $w \notin L$ y $w \notin V(T)$
            a. Concatene w al final de la lista L
            b. $V(T) = V(T) \cup \{w\}$ y $E(T) = E(T) \cup \{vw\}$



It	L	v	e
0	[a]	a	—
1	[c]	c	ac
2.	[b,e]	b,e	cb, ce
3.	[e]	—	—
4.	[d,f]	d,f	ed, ef
5.	[f]	—	—
6.	[g,h]	g,h	fg, fh
7.	[h]	—	—
8.	[ik]	i,k	hi, hk
9.	[k]	—	—
10.	[j]	j	kj
11.	[ ]	—	—





Altura : 6