



Práctica 7:

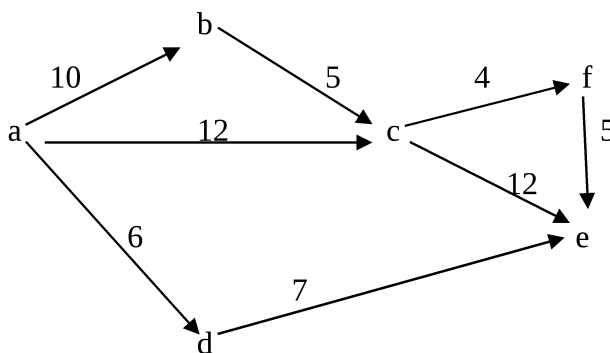
ÁRBOL DE EXTENSIÓN MINIMAL

OBJETIVO

- Cálculo del árbol de extensión (o generador) de un grafo.

PLANTEAMIENTO DEL PROBLEMA

En la empresa en la que trabajáis, ahora os han pedido que conectéis un grupo de ordenadores (llamados a,b,c,d,e,f) entre si para que se puedan compartir ficheros, el problema es que el cable que hay que usar es de alta velocidad y por tanto muy caro. Vuestro jefe ha dejado muy claro que como gastéis un metro de más os despide. Mediante el siguiente grafo se muestra qué ordenadores es posible conectar entre sí y cuantos metros tendría el cable. Vuestro objetivo por tanto consistirá en calcular el árbol de extensión (o árbol generador) minimal para dicho grafo.



a) El primer paso será implementar el predicado *arbol/2*, que calcula un árbol de extensión cualquiera. El algoritmo generará el árbol agregando en cada iteración un nuevo nodo a la solución parcial que lleve calculada. Para ello se llevará una lista con los nodos que faltan por añadir a la solución (L_n). Para añadir un nuevo nodo a la solución parcial, seleccionaremos un arco que tenga como nodo inicial (X) un nodo que no esté en L_n y como nodo final (Y) un nodo que sí esté en L_n . De esta forma habremos agregado el nodo Y a la solución parcial. A continuación se deberá borrar el nodo Y de la lista L_n y volveremos a realizar una nueva iteración hasta que la lista L_n quede vacía.

Al predicado *arbol/2* se le pasa como parámetro de entrada la lista L_n , que inicialmente contendrá todos los nodos menos el primero, y tendrá como resultado una lista con todos los arcos que forman el árbol de extensión.

Ejemplo de llamada al predicado *arbol/2*:

`arbol([b,c,d,e,f], L).` \rightarrow `L = [arco(a,b,10), arco(b,c,5), ...]`

Crear también un predicado *arbol/1*, que oculte la lista L_n al usuario por lo que su único parámetro será la lista resultado.



b) (Algoritmo de Prim) Implementar el predicado *arbol_min/2*, que deberá calcular el árbol de extensión minimal, es decir, de menor peso. Para ello modificaremos el predicado *arbol/2*, de manera que en vez de seleccionar un arco cualquiera en cada iteración, seleccionaremos el arco de peso mínimo que una un nodo de L_n con un nodo que no esté en L_n . Para calcular el arco mínimo utilizaremos el predicado *not*, de manera similar a la práctica anterior.