Tarea 4 Problemas conceptuales

Iván David Valderrama Corredor
Ingeniería de Sistemas y Ciencias de la Computación
Pontificia Universidad Javeriana, Cali

22 de marzo de 2019

Índice

1. Problemas conceptuales

1.1. Ejercicio 3.5:Binary Trees(Kleinberg y Tardos, página 108).

Un árbol binario es un árbol arraigado en el que cada nodo tiene como máximo dos hijos. Muestre por inducción que en cualquier árbol binario el número de nodos con dos hijos es exactamente uno menos que el número de hojas.

Respuesta

Caso Base:

Consideremos el primer nodo(raiz), el cual puede tener de 0 a 2 hojas.

-Si la raíz no tiene hijos entonces la raiz es una hoja y el árbol solo tiene 1 hoja, por lo que no tiene nodos con 2 hijos.
-Si la raíz tiene unicamente una hoja, entonces el arbol solo tiene 1 hoja y no tiene nodos con 2 hijos.
-Si la raíz tiene dos hojas, entonces el árbol tiene 2 hojas y cada nodo tiene 2 hijos.

Hipótesis inductiva:

$P(n)$: En un arbol binario con n nodos, el numero de nodos con dos hijos es exactamente $(n-1)$.

Supongamos que $P(n)$ es verdadero.

Caso1:
Tenemos un arbol binario con n+1 nodos, ahora agregaremos una hoja a un padre con solo un hijo.
Para este caso, el numero de nodos con dos hijos y hojas aumenta en 1

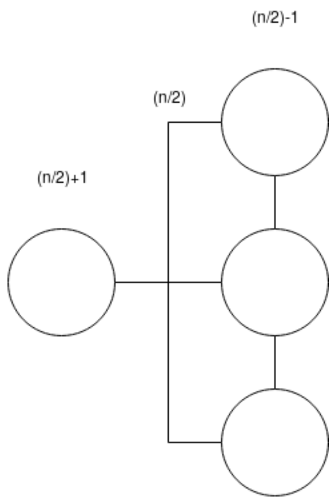
Caso2:
Tenemos un arbol binario con n+1 nodos, ahora agregaremos una hoja a un padre que es una hoja.
Para este caso, los padres con dos hijos y el numero de hojas siguen siendo iguales. [1]

1.2. Ejercicio 3.7:Wireless Networks(Kleinberg y Tardos, página 108).

Algunos de tus amigos trabajan en redes inalámbricas y actualmente están estudiando las propiedades de una red de dispositivos móviles. A medida que los dispositivos se mueven (en realidad, cuando sus dueños humanos se mueven), definen un gráfico en cualquier momento como sigue: hay un nodo que representa cada uno de los n dispositivos, y hay un borde entre el dispositivo i y el dispositivo j si el dispositivo físico Las ubicaciones de i y j no están a más de 500 metros de distancia. (si es así, digamos que i y j) están "dentro del rango.[el] uno del otro.)
Les gustaría que la red de dispositivos esté conectada en todo momento, por lo que han restringido el movimiento de los dispositivos para satisfacer los siguientes requisitos: en todo momento, cada dispositivo está a 500 metros de al menos n / 2 de los otros dispositivos. (Asumiremos que n es un número par). Lo que les gustaría saber es: ¿Esta propiedad por sí misma garantiza que la red permanecerá conectada?
Decida si cree que la afirmación es verdadera o falsa, y proporcione una prueba de la reclamación o su negación.

Respuesta

Supongamos que tenemos un grafo G(n,v)
En la primera capa del grafo tenemos $(n/2)+1$ canditad de vertices, en la segunda capa del grafo tenemos $(n/2)-1$ cantidad de vertices y asi sucesivamente hasta n. Cuando la cantidad de nodos es negativa, se empieza a reutilizar los nodos previamente empleados, enlazando caminos con dichos nodos, lo que prueba que el la red permanecera conectada, pues este grafo es un grafo conexo.



1.3. Ejercicio 4.2:Minimum Spanning Trees(Kleinberg y Tardos, página 189).

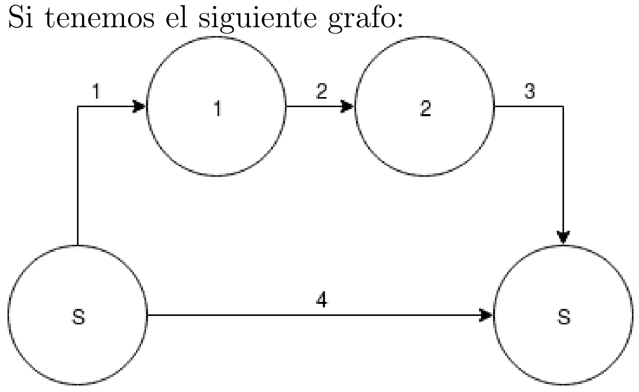
Para cada una de las siguientes dos afirmaciones, decida si es verdadero o falso. Si es verdad, da una breve explicación. Si es falso, da un contraejemplo.

(a) Suponga que se nos da una instancia del Problema del árbol de expansión mínima en un grafo G, con costos de borde que son todos positivos y distintos. Sea T un árbol de expansión mínimo para esta instancia. Ahora supongamos que reemplazamos cada costo de borde por su cuadrado, c_e^2. creando así una nueva instancia del problema con el mismo grafo pero con diferentes costos. ¿Verdadero o falso? T debe ser un árbol de expansión mínimo para esta nueva instancia.

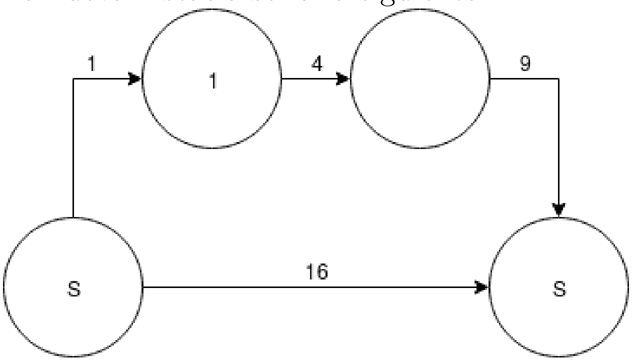
Respuesta
Dado un arreglo de costos de arcos ordenados de manera creciente.
$A = [c_1, c_2, c_3...c_{n-1}]$
se cumple la siguiente propiedad
si $c_1 < c_2$ entonces $(c_1)^2 < (c_2)^2$
y esto se cumple para cualquier c_n
Ahora debido a que el orden del arreglo no es afectado al momento de reemplazar los costos por el cuadrado de los mismos, el arbol de expansion minima no cambia, por lo que el enunciado es verdadero.

(b) Supongamos que se nos da una instancia del problema de ruta s-t más corta en un grafo dirigido G. Suponemos que todos los costos de borde son positivos y distintos. Sea P una ruta s-t de costo mínimo para esta instancia. Ahora supongamos que reemplazamos cada costo de borde por su cuadrado, c_e^2, creando así una nueva instancia del problema con el mismo grafo pero con diferentes costos. ¿Verdadero o falso? P debe seguir siendo una ruta s-t de costo mínimo para esta nueva instancia.

Respuesta
Si tenemos el siguiente grafo:



La nueva instacia seria la siguiente:



Como podemos ver en la primera instancia del grafo, el camino empleado es el de (S,F)4=4. <(S,1,2,F)1+2+3=6
En cambio en la segunda instancia del grafo, el camino empleado es el (S,1,2,F)1+4+9=14 <(S,F)16=16.

1.4. Ejercicio 4.5:Quiet Country Roads(Kleinberg y Tardos, página 191).

Consideremos un camino rural largo y tranquilo con casas dispersas muy dispersas a lo largo. (Podemos imaginar la carretera como un segmento de línea larga, con un punto final oriental y un punto final occidental).
Además, supongamos que a pesar de la configuración bucólica, los residentes de todas estas casas son ávidos usuarios de teléfonos celulares. Desea colocar estaciones base de teléfonos celulares en ciertos puntos a lo largo de la carretera, de modo que cada casa esté a cuatro millas de una de las estaciones base.
Proporcione un algoritmo eficiente que logre este objetivo, utilizando la menor cantidad de estaciones base posibles

Respuesta

Entrada: Un arreglo P[0..N] de numeros naturales, los cuales representan las posiciones de las casas
Salida: Numero Natural A que representa la cantidad minima de estaciones base empleadas, tal que cubran 4 millas de distancia por casa.

```
def QCR(A):
    A.sort()
    ans,low,n,ok,N = 0,A[0],True,len(A)
    while ok and low<A[len(A)-1] and n!=N:
```

```
    ok = A[n]<=low
    best,n = n,n+1
    while ok and n!=N and A[n][0]<=low:
      if A[n]>A[best]:
        best = n
      n += 1
    ans+=best
    low = A[best]
  ok = ok and low>=A[len(A)-1]
  if ok==False:
    ans = 0
  return ans
```

[2]

Referencias

[1] Course Hero, *Problem 9 a binary tree is a rooted tree in which.* https://www.coursehero.com/file/p5sovqg/Problem-9-A-binary-tree-is-a-rooted-tree-in-which-each-node-has-at-most-two/

[2] Camilo Rocha , *Minimal Covering Algorithm with failure checking.* https://https://bitbucket.org/snippets/hquilo/z6KbA