Búsqueda en amplitud

Guillermo Palma

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

Plan

Preliminares

Camino de costo mínimo (shortest-path distance)

Definición

Dado un grafo $G = (V, E)$, en donde todos los lados tiene costo 1. Definimos un **camino de costo mínimo** desde un vértice s hasta un vértice v, como el camino de menor número de lados en G desde s hasta v. Lo denotamos como $\delta(s, v)$. Si no hay camino entre s y v en G, decimos que $\delta(s, v) = \infty$.

Propiedad de los caminos de costo mínimo

Lema 1

Sea $G = (V, E)$ un digrafo o un grafo no dirigido, sea s un vértice cualquiera de V, y sea $(u, v) \in E$, entonces se cumple que:

$$\delta(s, v) \leq \delta(s, u) + 1$$

Prueba:

Si el vértice u es alcanzable desde s, entonces v también lo es. Entonces, el camino $\delta(s, u)$ o pasa por v, o se alcanza a v con el lado (u, v). En consecuencia, la desigualdad se cumple. Si el vértice u no es alcanzable desde s, entonces $\delta(s, u) = \infty$. En consecuencia, la desigualdad se cumple. $\square$

Propiedades de los árboles

Teorema B.2 de [1] (Properties of free trees)

Sea $G = (V, E)$ un grafo no dirigido. Las siguientes afimaciones son equivalentes.

- G es un árbol (*free tree*).
- Cualquier par de vértices de G están conectados por un camino simple que es único.
- G es conectado, y $|E| = |V| - 1$.
- G es acíclico, y $|E| = |V| - 1$.
- G es acíclico, pero si se agrega un lado a E, entonces el grafo resultante va a tener un ciclo.

Algoritmo de búsqueda en amplitud

Sobre la búsqueda en amplitud o Breadth-first search (BFS)

- Se quiere recorrer un grafo $G = (V, E)$ desde un vértice $s \in V$, hasta cada uno de sus vértices alcanzables en V.
- Suponemos que cada lado $G = (V, E)$ tiene un costo de 1.
- Se computa la distancia entre s y cada uno de los vértices alcanzables.
- La distancia computada corresponde al camino de costo mínimo.
- El grafo asociado con el resultado de BFS, que tiene como raíz a s y cada uno de los caminos desde s hasta sus vértices alcanzables es un árbol.
- Funciona en digrafos y grafos no dirigidos.
- El algoritmo alcanza primero a todos los vértices que están a una distancia k antes de alcanzar a los que están a una distancia de $k + 1$.
- Se va aplicar desde un vértice fijo s, pero puede ser modificado para ser aplicado a todos los vértices.

Algoritmo de búsqueda en amplitud

Procedimiento BFS(G =(V, E), s)

inicio
  para cada $v \in V$ **hacer**
    $v.d \leftarrow \infty$; $v.color \leftarrow BLANCO$; $v.pred \leftarrow NIL$;
  $s.d \leftarrow 0$; $s.color \leftarrow GRIS$; $Q \leftarrow \emptyset$;
  $Q.encolar(s)$;
  mientras $Q \neq \emptyset$ **hacer**
    $u \leftarrow Q.desencolar()$;
    para cada $v \in G.adyacente[u]$ **hacer**
      si $v.color = BLANCO$ **entonces**
        $v.color \leftarrow GRIS$;
        $v.d \leftarrow u.d + 1$;
        $v.pred \leftarrow u$;
        $Q.encolar(v)$
    $u.color \leftarrow NEGRO$

Ejemplo de la búsqueda en amplitud

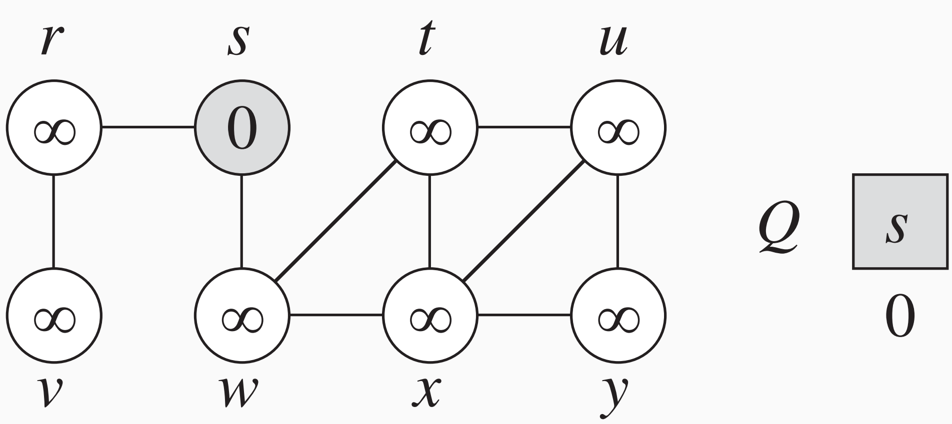

Figura 1: Comienzo de BFS, se encola el vértice inicial s. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

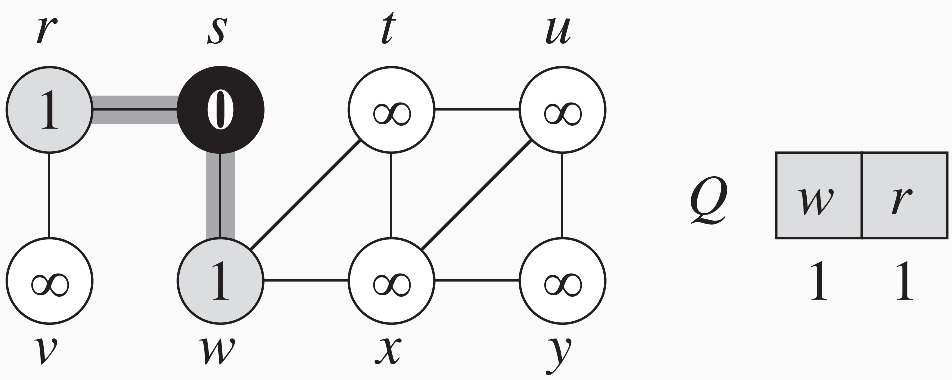

Figura 2: Se desencola el vértice s y se encolan los vértices w y r, que son adyacentes a s. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

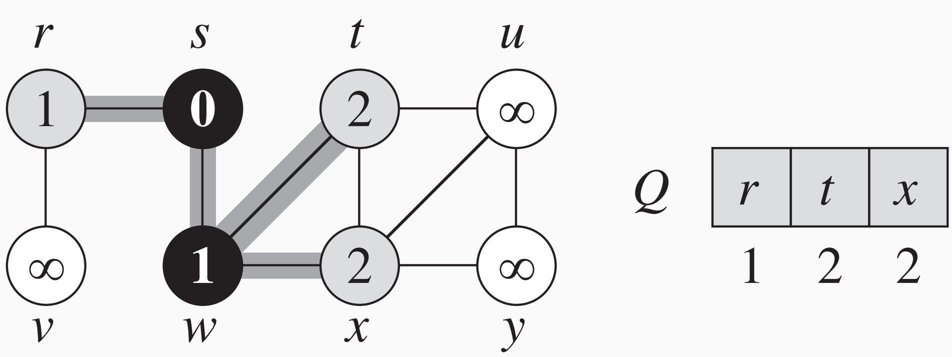

Figura 3: Se desencola el vértice w y se encolan los vértices t y x, que son adyacentes a w. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

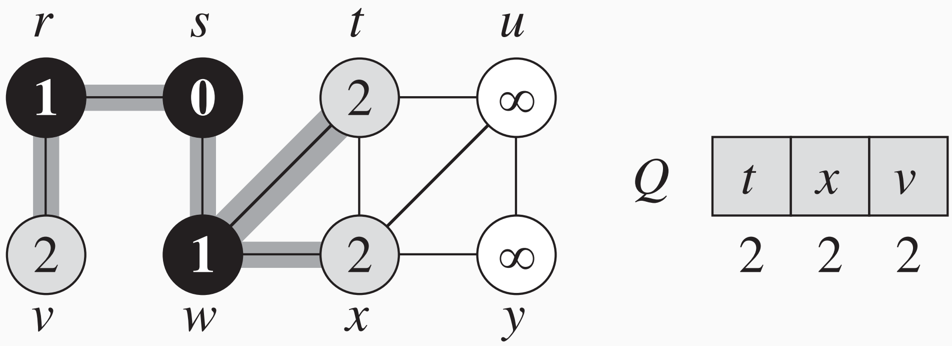

Figura 4: Se desencola el vértice r y se encola el vértice v, que es adyacentes a r. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

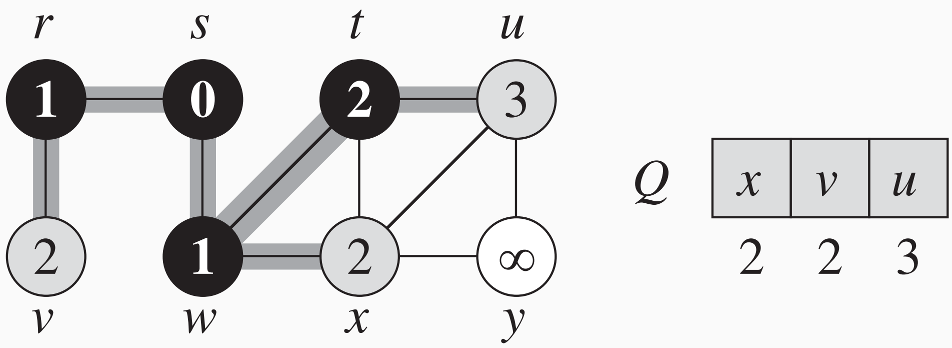

Figura 5: Se desencola el vértice t y se encola el vértice u, que es adyacentes a t. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

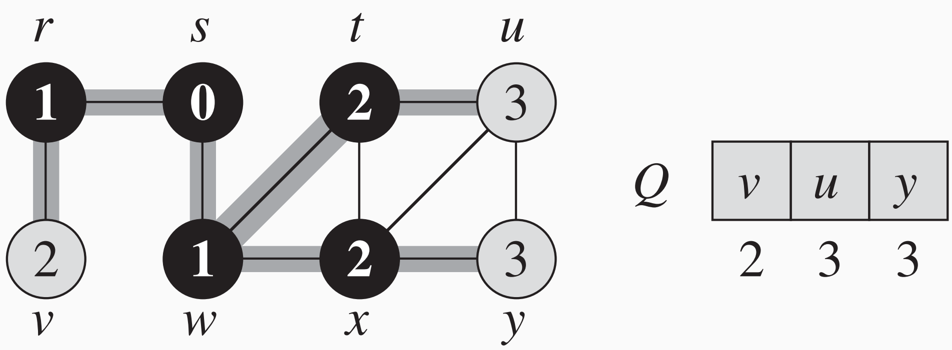

Figura 6: Se desencola el vértice x y se encola el vértice y, que es adyacentes a x. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

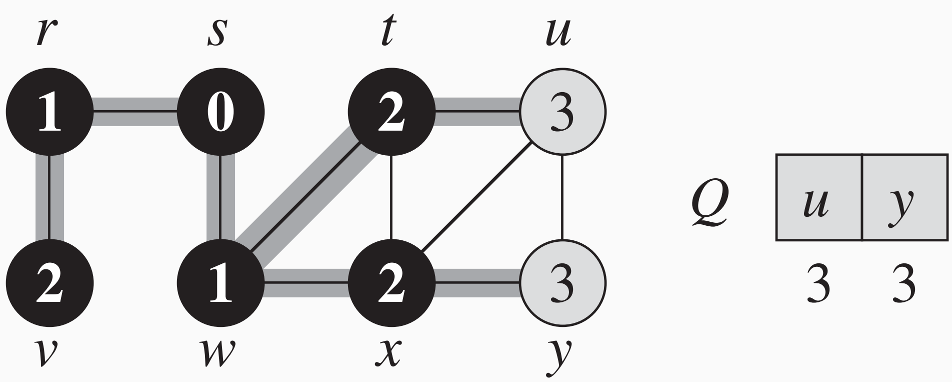

Figura 7: Se desencola el vértice v. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

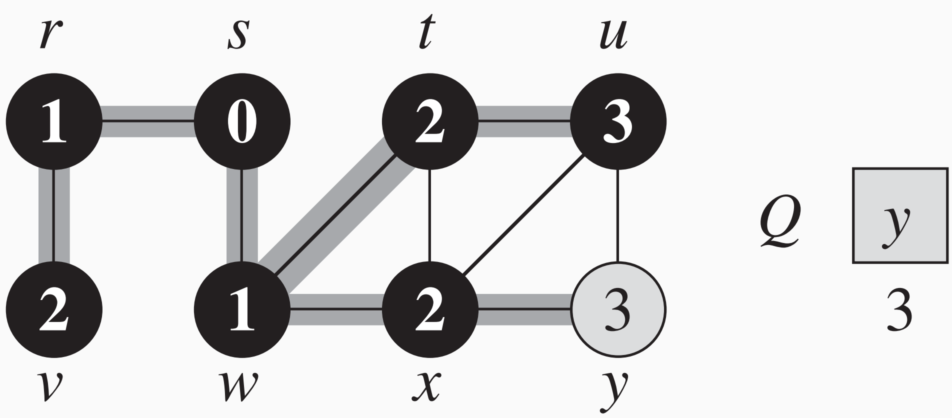

Figura 8: Se desencola el vértice u. Fuente [1].

Ejemplo de la búsqueda en amplitud, continuación

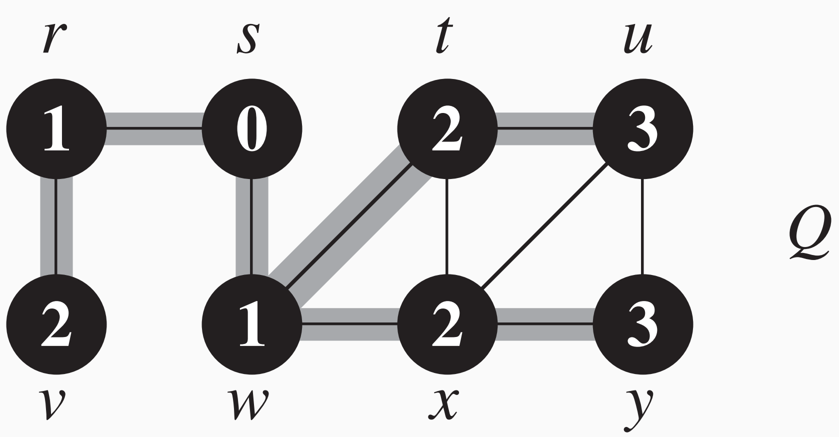

Figura 9: Se desencola el vértice y, la cola queda vacía y se termina la ejecución BFS. Fuente [1].

Análisis del tiempo de la búsqueda en amplitud

- Un vértice entra en la cola si esta marcado de blanco.
- Observe que a todos los vértices se marcan de blanco una sola vez al comienzo.
- Todo vértice que entra la cola sale de ella y como solo entra una vez las operaciones de la cola son $O(|V|)$.
- A cada vértice que sale de la cola se le obtiene los vértices adyacentes.
- Si a todos los vértices que salen de la cola se le obtiene los adyacentes entonces, las operaciones del bucle *mientras* es $\Theta(|E|)$.
- Se inicializan todos los vértices al comienzo del algoritmo, esto es $O(|V|)$.
- Por lo tanto, el tiempo de BFS es $O(|V| + |E|)$.

Propiedades de la búsqueda en amplitud

Distancia de cada vértice alcanzado después de BFS

Lema 2

Sea $G = (V, E)$ un digrafo o un grafo no dirigido, y suponga que se ejecuta BFS en G desde un vértice $s \in V$. Entonces, cuando BFS finaliza se tiene que la distancia computada por BFS para cada vértice $v \in V$, satisface $v.d \geq \delta(s, v)$

Distancia de cada vértice alcanzado después de BFS, cont.

Prueba Lema 2

Por inducción.

- **H.I.**: $v.d \geq \delta(s, v)$ esto es $\forall v \in V$
- **Caso Base**: se cumple, por encolar s línea 5 $s.d = 0 = \delta(s, s)$ y $v.d = \infty \geq \delta(s, v)$ esto es $\forall v \in V - \{s\}$.
- **Caso Inductivo**: sea v un vértice en blanco alcanzado desde u. Por lema 1 y línea 11, se tiene que:

$$v.d = u.d + 1$$
$$v.d \geq \delta(s, u) + 1$$
$$v.d \geq \delta(s, v)$$

- Una vez que v es encolado, se pinta de gris y por lo que que no se encola de nuevo. Por lo tanto, $v.d$ no cambia y la H.I. se cumple. $\square$

Distancia entre los elementos de una cola de BFS

Lema 3

Suponga que se está en ejecutando BFS en grafo G y se tiene la cola $Q = \langle v_1, v_2, v_3, \ldots, v_r \rangle$, tal que v_1 es la cabeza de Q y v_r la cola. Entonces, se cumple que:

1. $v_r.d \leq v_1.d + 1$ y
2. $v_i.d \leq v_{i+1}.d$,

para $i = 1, 2, 3, \ldots, r - 1$.

Distancia entre los elementos de una cola de BFS, cont.

Prueba Lema 3

- Por inducción sobre los encolamientos/des-encolamientos.
- **Caso base**: Se cumple cuando la cola contiene solo a s.
- **Caso Inductivo**: Se quiere probar 1) y 2) se cumplen después de cada operación de encolamiento/des-encolamiento.
- Caso se desencola un elemento en línea 7.
  - Si se desencola v_1, entonces v_2 es la cabeza de la lista.
  - $v_1.d \leq v_2.d$ por H.I. en 2).
  - De 1) se tiene $v_r.d \leq v_1.d + 1 \leq v_2.d + 1$.
  - Entonces si se desencola v_1 el lema se cumple con v_2 como cabeza.
- Caso se encola un elemento en línea 13.
  - Cuando se encola un elemento este es v_{r+1}.
  - Se desencoló u y la nueva cabeza es v_1, entonces $u.d \leq v_1.d$.
  - Entonces, $v_{r+1}.d = v.d = u.d + 1 \leq v_1.d + 1$.
  - Por H.I. $v_r.d \leq u.d + 1$, entonces $v_r.d \leq u.d + 1 = v.d = v_{r+1}.d$.
  - Entonces el Lema 3 se cumple cuando v es encolado. $\square$

Distancia entre dos vértices encolados de BFS

Corolario 1

Suponga que v_i y v_j son dos vértices encolados durante la ejecución de BFS y suponga que v_i fue encolado antes que v_j. Entonces, se tiene que se cumple que $v_i.d \leq v_j.d$ en el momento que v_j es encolado.

Prueba

Por Lema 3 y por el hecho de que la distancia de un vértice durante la ejecución de BFS se modifica solo una vez cuando es encolado. $\square$

Correctitud del algoritmo búsqueda en amplitud

Teorema 1

Sea $G = (V, E)$ un digrafo o un grafo no dirigido y en donde se ejecuta BFS desde el vértice s, tal que $s \in V$. Durante la ejecución de BFS, se obtienen todos los vértices $v \in V$ que son alcanzables desde s, y después de su terminación se tiene que $v.d = \delta(s, v)$ para todo $v \in V$. Además para cualquier vértice $v \neq s$ que es alcanzable desde s, se tiene que el camino de costo mínimo desde s hasta v es igual al camino de costo mínimo desde s hasta $v.pred$ seguido del lado $(v.pred, v)$.

Prueba Teorema 1

- Por contradicción. Se tiene un vértice v y $v.d \neq \delta(s, v)$.
- $v \neq s$ y v es alcanzable desde s, sino $\delta(s, v) = \infty \geq v.d$.
- Por lema 2 $v.d \geq \delta(s, v)$, entonces $v.d > \delta(s, v)$.
- Sea u el vértices predecesor de v, entonces $\delta(s, v) = \delta(s, u) + 1$.
- Se tiene que $\delta(s, v) > \delta(s, u)$ y $\delta(s, u) = u.d$.
- $v.d > \delta(s, v) = \delta(s, u) + 1 = u.d + 1 \quad (1)$.
- Suponemos que se desencola u en línea 7.
- Si v es blanco, entonces $v.d = u.d + 1$ lo que contradice ec. (1).
- Si v es negro, entonces $v.d \leq u.d$ lo que contradice ec. (1).
- Si v es gris entonces se desencoló un w antes que u y $v.d = w.d + 1$.
- Por Corolario 4 $w.d \leq u.d$, entonces $v.d = w.d + 1 \leq u.d + 1$ lo que contradice ec. (1).
- Entonces se concluye que $v.d = \delta(s, v)$.
- Se tiene que $v.pred = u$, entonces $v.d = u.d + 1$, por lo tanto se pueden obtener los caminos de costo mínimo desde s. $\square$

Subgrafo predecesor

Definición

Sea $G = (V, E)$ un digrafo o un grafo no dirigido con vértice fuente s, tal que $s \in V$. Se define el **subgrafo predecesor** como:
$G_{pred} = (V_{pred}, E_{pred})$ donde: $V_{pred} = \{v \in V : v.pred \neq NIL\} \cup \{s\}$ y $E_{pred} = \{(v.pred, v) : v \in V_{pred} - \{s\}\}$.

Definición

El subgrafo predecesor G_{pred} es un **árbol de BFS** (*breadth-first tree*) si V_{pred} consiste en todos los vértices alcanzables desde s y para todo $v \in V_{pred}$, existe un camino simple de costo mínimo desde s hasta v.

Subgrafo predecesor resultante de aplicar BFS

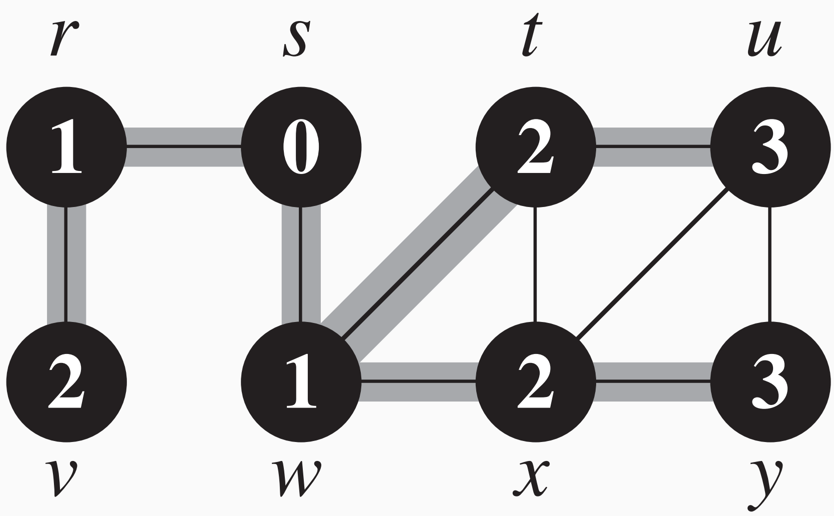

Figura 10: *Breadth-first tree* que se obtiene de la ejecución BFS. Fuente [1].

Subgrafo predecesor generado por BFS

Lema 4

Sea $G = (V, E)$ un digrafo o un grafo no dirigido tal que se le aplica BFS, se tiene que BFS genera predecesores en los vértices tal que el $G_{pred} = (V_{pred}, E_{pred})$ asociado, es un árbol de BFS (*breadth-first tree*).

Subgrafo predecesor generado por BFS, cont.

Prueba Lema 4

- En línea 12 de BFS se tiene que $v.pred \leftarrow u$.
- Línea 12 ocurre si solo si $(u, v) \in E$ y si v es alcanzable desde s.
- Entonces, V_{pred} consiste de vértices que son alcanzables desde s.
- Como G_{pred} forma un árbol, entonces por Teorema B2, hay un camino simple único entre s y los vértices de V_{pred}.
- Aplicando el Teorema 1 inductivamente, entonces cada camino entre s y los vértices de V_{pred}, es un camino de costo mínimo de G. $\square$

Imprimiendo el camino de costo mínimo de BFS

Procedimiento imprimir-camino(G =(V, E), s, v)

```
inicio
    si v = s entonces
        imprimir(s) ;
    si no, si v = NIL entonces
        imprimir(No existe un camino desde s hasta v) ;
    en otro caso
        imprimir-camino(G, s, v.pred) ;
        imprimir(v)
```

Referencias

[1] T. Cormen, C. Leirserson, R. Rivest, and C. Stein.
Introduction to Algorithms.
McGraw Hill, 3ra edition, 2009.