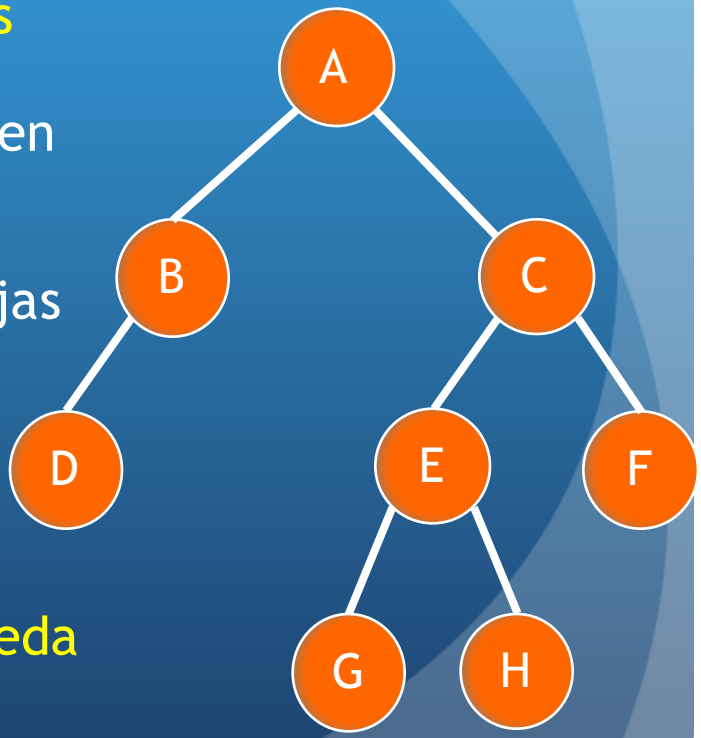
Estructuras de Datos IIS262

Profesor: Patricio Galeas

CAPÍTULO 9 : Árboles RojoNegro

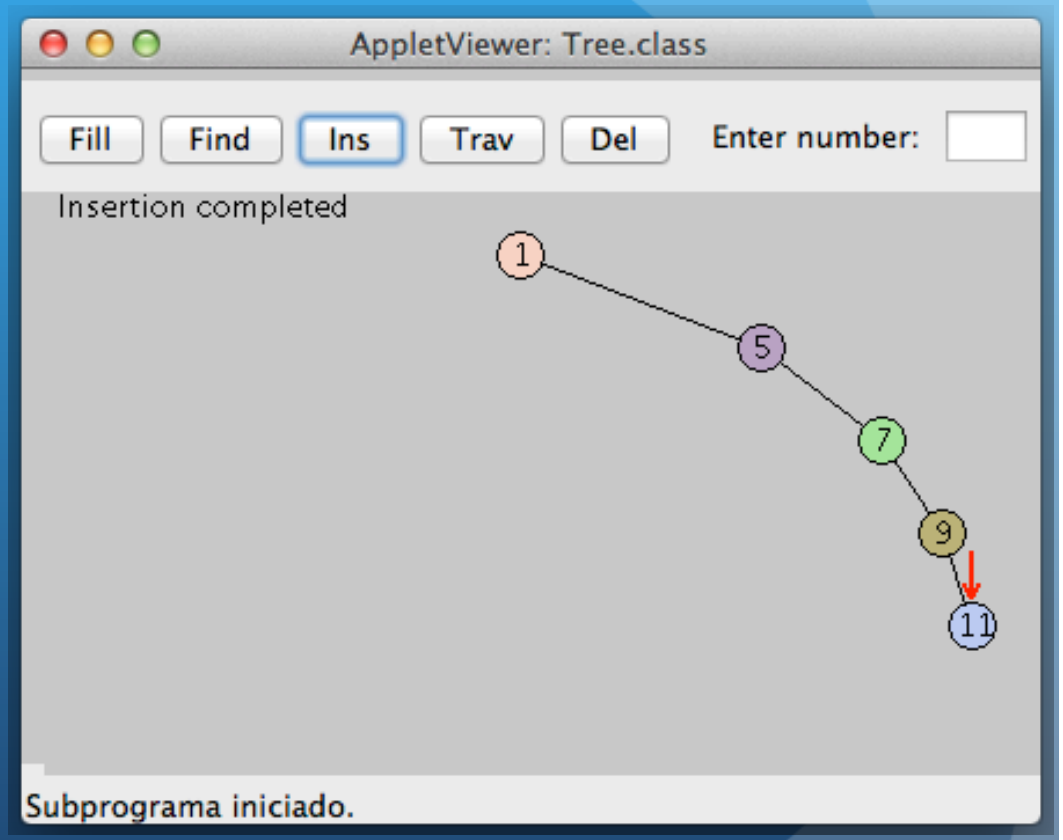
Introducción

- Ya vimos que los árboles de búsqueda binarios poseen grandes ventajas como estructuras de almacenamiento: búsqueda, inserción y eliminación rápida.
- Desventaja: funcionan bien cuando los datos son ingresados en forma aleatoria, pero son muy lentos cuando los datos son ingresados en forma ordenada.
- Árbol desbalanceado pierde todas sus ventajas de búsqueda, inserción y eliminación.
- Ahora veremos la forma de resolver este problema: árboles rojo-negro.
- Los árboles rojo-negro son árboles de búsqueda binario con algunas "características especiales".



Arboles balanceados y desbalanceados

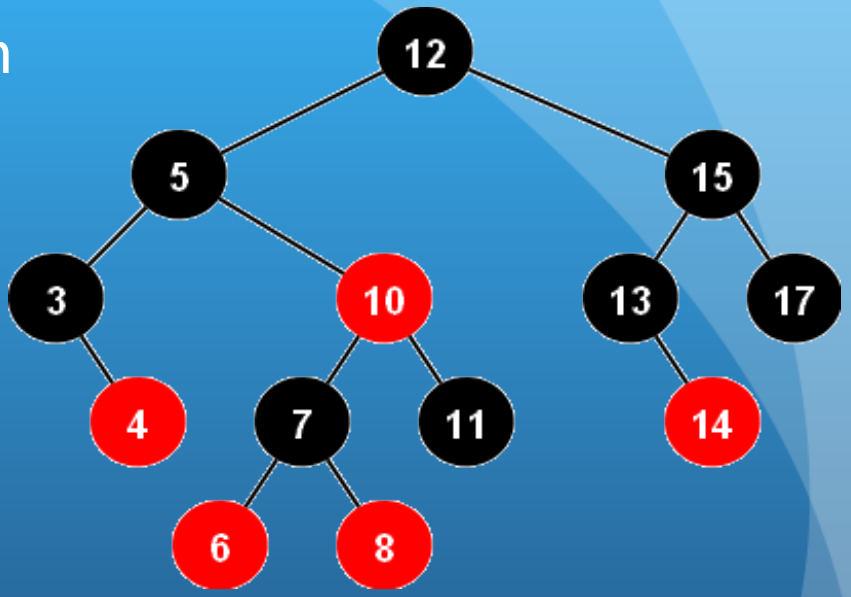
- Si ingresamos nodos con valores ascendentes en el applet Tree.
- Se genera un árbol extremadamente desbalanceado: como una lista enlazada.
- La eficiencia se transforma:
  O(log N) -> O(N)
- Si generamos arboles con pocos valores aleatorios -> árbol parcialmente balanceado -> degrada eficiencia en la búsqueda.



Arboles balanceados y desbalanceados

- Para garantizar una búsqueda en tiempo O(log N), debemos asegurarnos que el árbol esté siempre balanceado.
- Balanceado :
  En cada nodo del árbol …

  #nodos izq ≅ #nodos der.
- En los árboles Rojo-Negro (RB) se intenta mantener el balance durante la inserción y el borrado de nodos.
- Cuando no se logra, se aplica una rutina de re-estructuración.



Características de los árboles RB

- Hay dos características en los árboles RB, una simple y otra un poco más compleja
  - Los nodos poseen colores.
  - Hay ciertas REGLAS para la inserción y eliminación para conservar la distribución de colores.

Reglas de los árboles RB

1. Cada nodo puede ser rojo o negro
2. El nodo raíz es siempre negro
3. Si un nodo es rojo, sus hijos deben ser siempre negros.
4. Todas las rutas desde la raíz a las hojas, o a un hijo nulo, tiene la misma cantidad de nodos negros (altura negra).

- **El Hijo Nulo** : es el lugar donde un nodo puede ser insertado a otro nodo (que no es una hoja).
  En otras palabras:
  - Es el potencial hijo izquierdo de un nodo con un solo hijo derecho.
  - Es el potencial hijo derecho de un nodo con un hijo izquierdo.
- **Altura Negra** : Es el número de nodos negros desde el raíz hasta una hoja.

¿Que hacer cuando las reglas de los arboles RB son violadas?

Solo hay dos posibilidades

1. Se puede cambiar el color de los nodos.
2. Se pueden hacer ciertas rotaciones.

RBTree

applet

Experimentos con el Applet

Experimento #1 : Insertando 2 nodos rojos

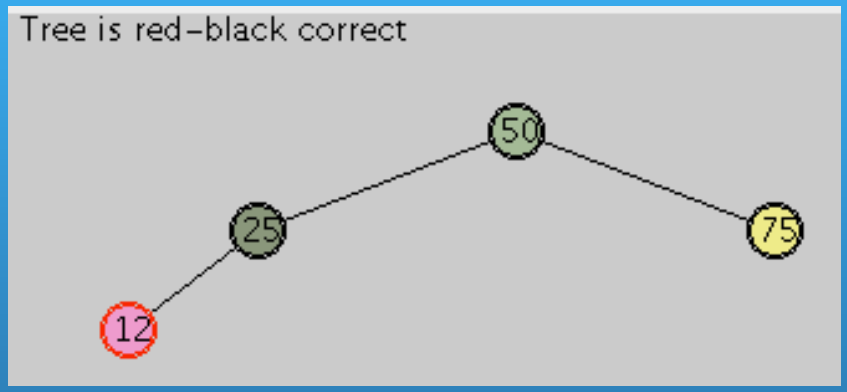
Experimento #2 : Rotaciones

Experimento #3 : Intercambiando colores

RBTree

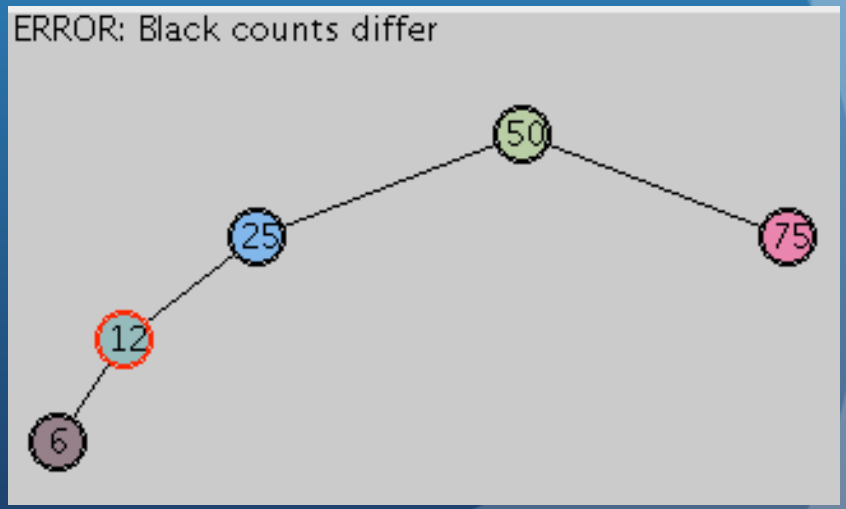
applet

Árbol Balanceado



A pesar de que no esta perfectamente balanceado, no alcanza a violar ninguna de las 4 reglas.

Árbol Desbalanceado



Cuando una de las rutas del árbol difiere de otra en dos o mas niveles, alguna de las 4 reglas estará siendo necesariamente violada.

… entonces hay que re-balancear el árbol con cambios de color y rotaciones.

Seguir la 4 reglas garantiza un árbol balanceado

Traten de crear un árbol desbalanceado en dos o más niveles que también cumpla con las 4 reglas de un árbol Rojo/Negro.

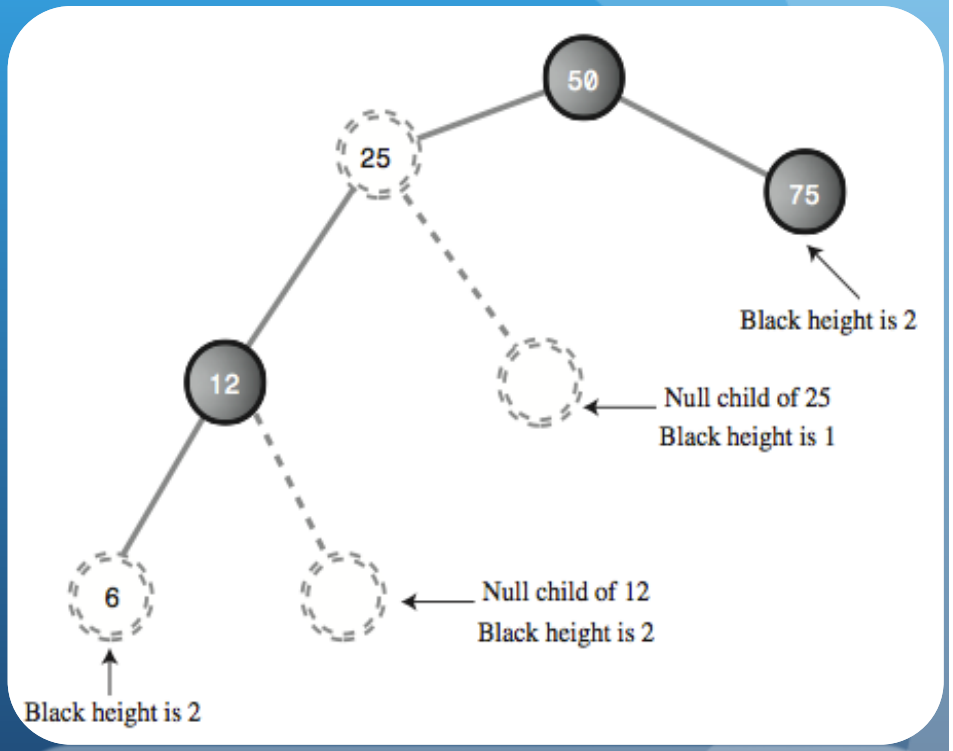
¿Es posible?

Hijo Nulo

Recordemos que la regla 4 especifica que todos las rutas que van desde la raíz hasta una hoja o hasta cualquier **hijo nulo** debe tener la misma cantidad de nodos negros.

Un hijo nulo es un hijo que un nodo (que no es hoja) podría tener, pero que no lo tiene.

Ejemplo: es el hijo izquierdo inexistente de un nodo que tiene un solo hijo derecho.



Rotaciones y Balanceo

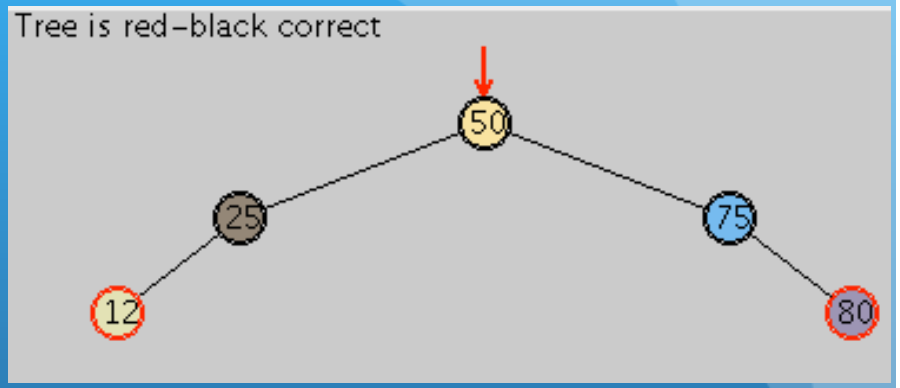
Para balancear un árbol tenemos que ejecutar algunas rotaciones

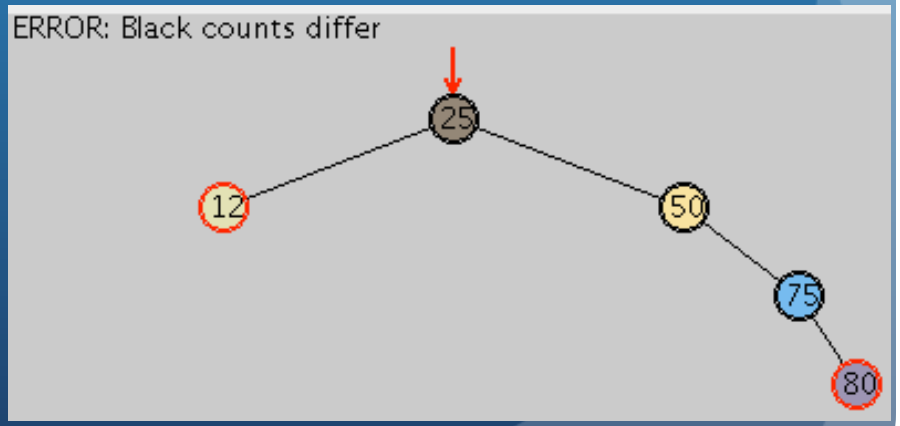
Por ejemplo: si todos los nodos están a la izquierda de la raíz, hay que mover algunos nodos al lado derecho, ***preocupándose de que las características de los árboles binarios se siga cumpliendo***.

OBS: En general las acciones de cambio de color de los nodos sólo sirve para decidir cuando hacer una rotación.

Rotaciones y Balanceo

- En la rotación sólo se cambia la relación entre los nodos.
- Para esto, se tiene que elegir un nodo como referencia "**top** de la rotación".
- Si rotamos a la derecha, el top de la rotación toma la posición de su hijo derecho y el hijo izquierdo toma la posición del padre.

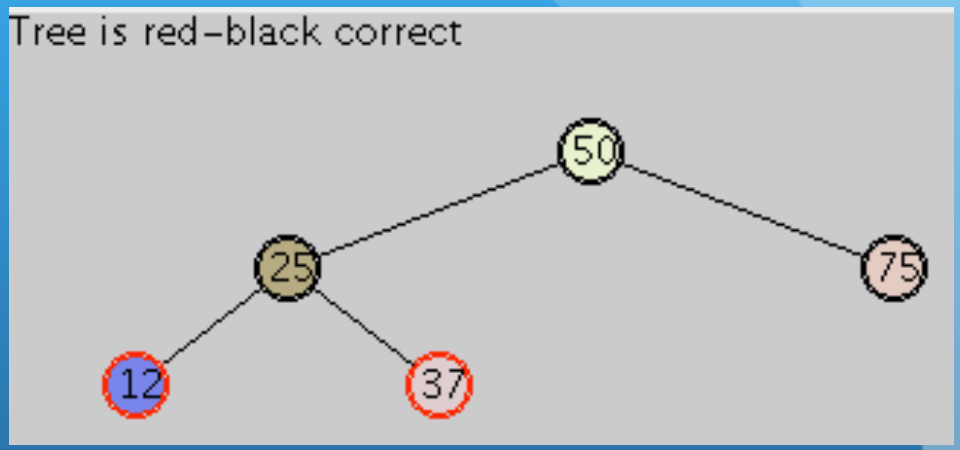


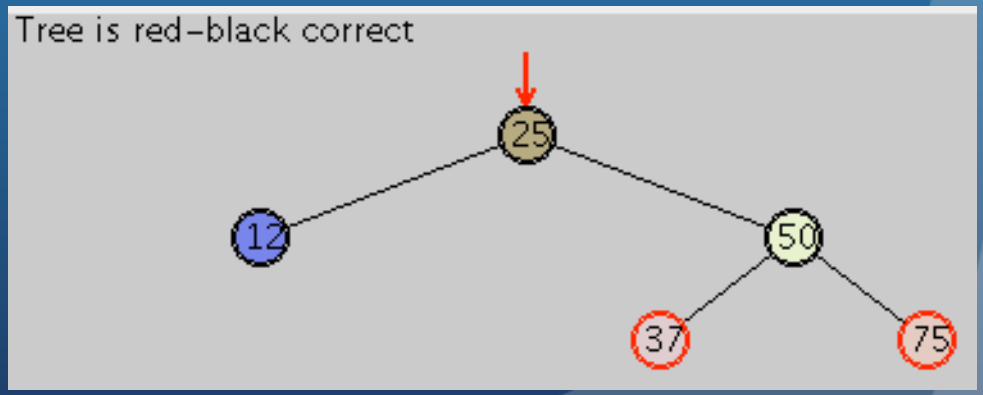


Rotaciones y Balanceo

Extraño salto de un nodo!

- En la rotación puede ocurrir que un nodo cambie de padre como ocurre en este ejemplo
- El nodo 37 pasa a ser el hijo izquierdo de su abuelo.

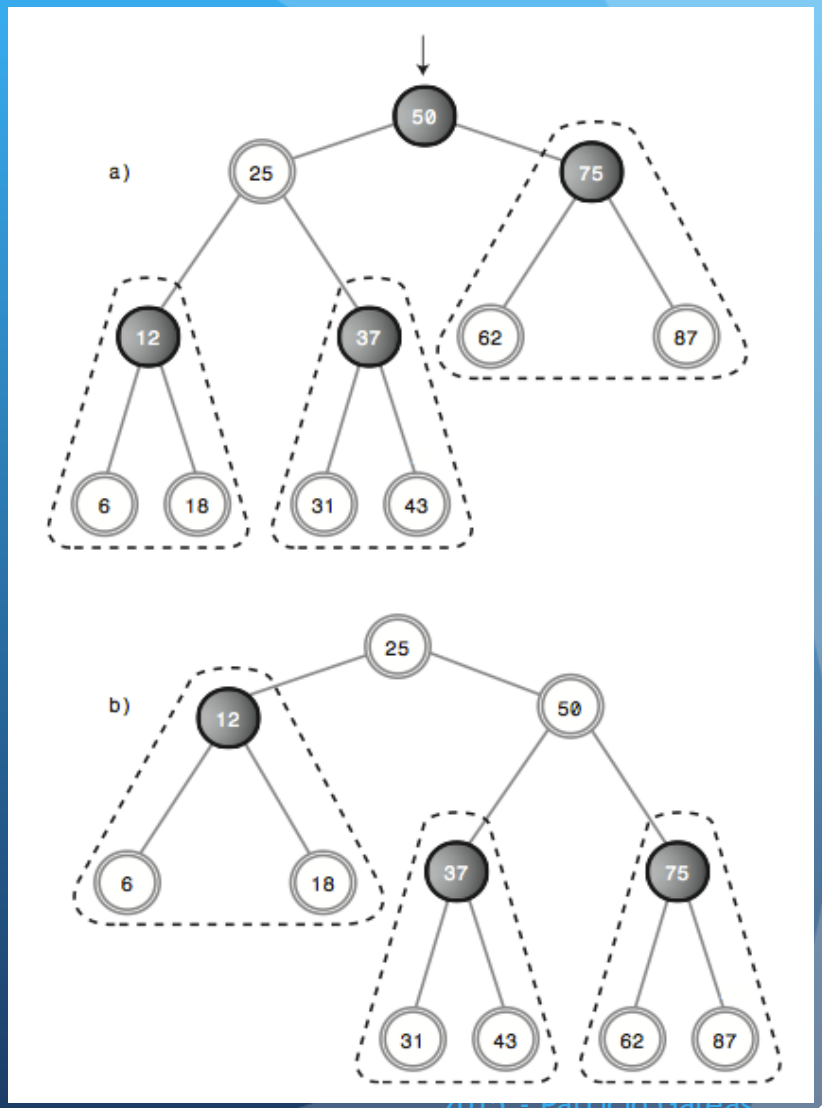




Rotaciones y Balanceo

Lo mismo puede ocurre con los sub-árboles!

- En la rotación puede ocurrir que un sub-árbol cambie de padre como ocurre en este ejemplo
- El sub-árbol de 37 pasa a ser el hijo izquierdo del nodo 50.



Seres Humanos vs. Computadores

Los seres humanos podemos ejecutar las rotaciones necesaria para balancear un árbol solo “observando” el la estructura/patrón actual del árbol.

… los computadores lamentable- /afortunadamente … no pueden hacer lo mismo y tienen que seguir ciertas reglas.

Insertando un nuevo Nodo

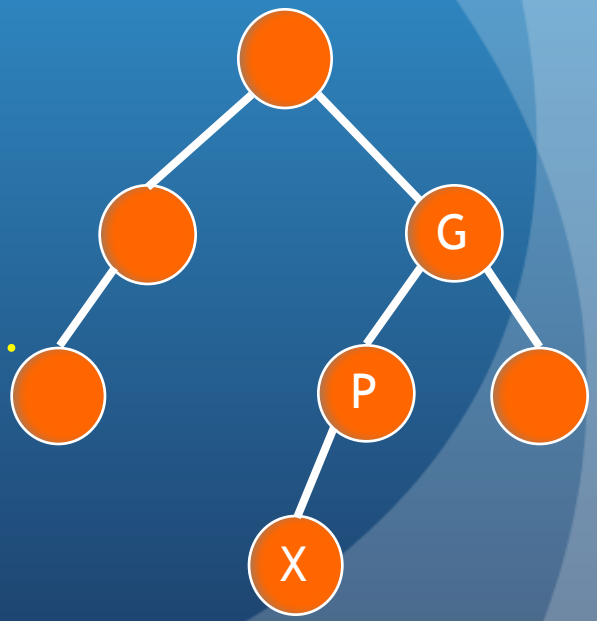
El nodo que se inserta es siempre de color **ROJO**.

Para el análisis tendremos la siguiente estructura:

- **X** : es un nodo particular que causa la violación de una regla.
- **P** : es el padre de X
- **G** : es el abuelo de X

La inserción del nodo se analizara en tres partes ordenadas según su complejidad:

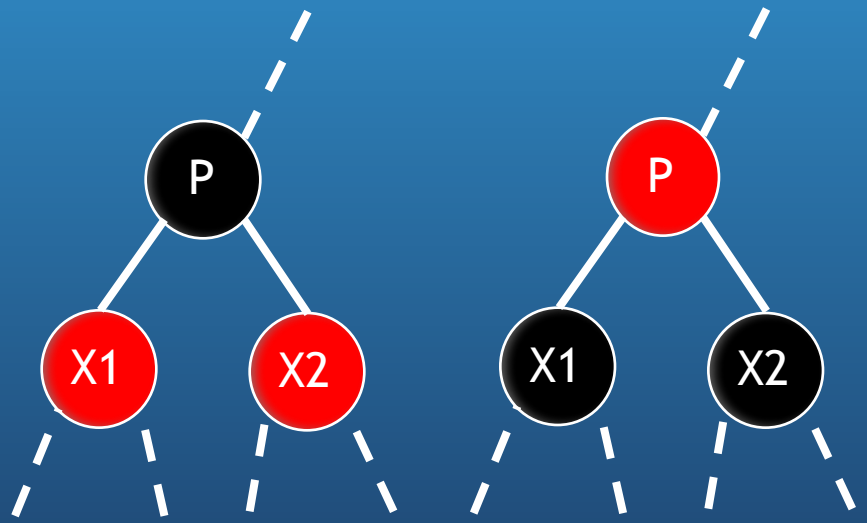
1. Cambios de color en el recorrido hacia abajo.
2. Rotaciones después de la inserción del nodo.
3. Rotaciones en el recorrido hacia abajo.



Insertando un nuevo Nodo

1. Cambio de color en el recorrido hacia abajo

Regla:
Cada vez que la rutina de inserción encuentre un nodo negro que tiene 2 hijos rojos, esta tiene que cambiar a los 2 hijos a negro y al padre a rojo. A menos que el padre sea la raíz.

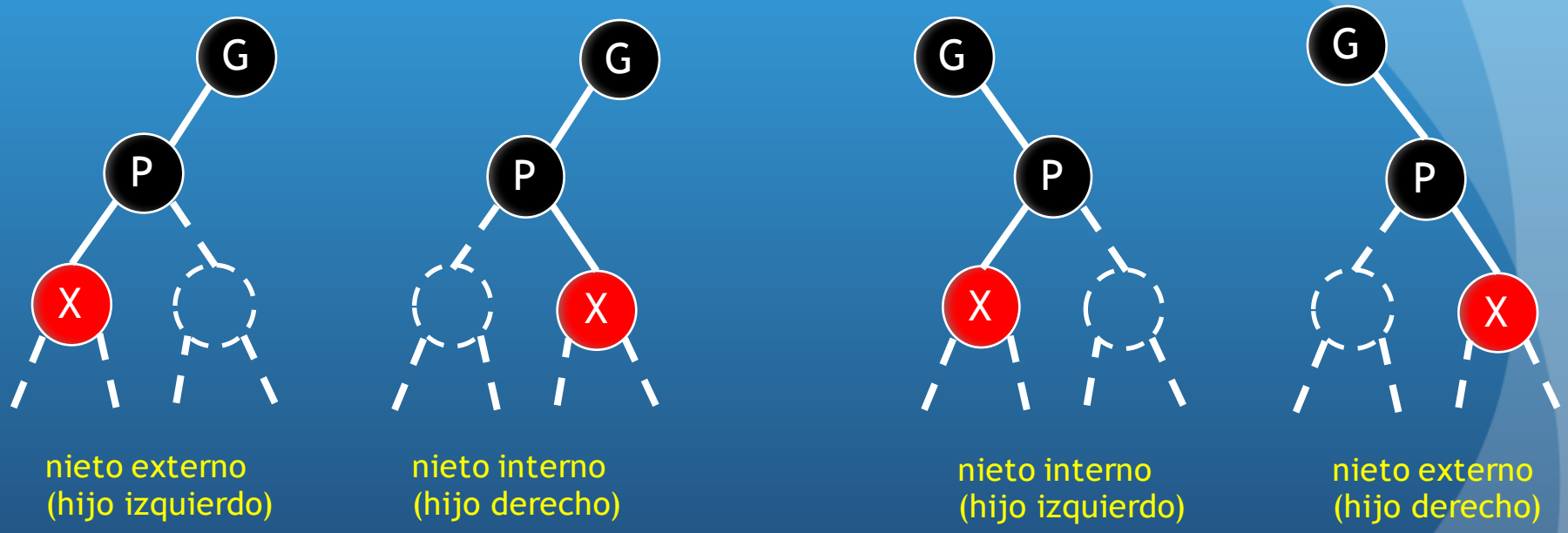


- No genera una variación en la cantidad de nodos negros en las rutas: no viola regla 4.
- El cambio permite agregar nuevos nodos debajo de X1 y X2.

Insertando un nuevo Nodo

2. Rotaciones después de insertar el Nodo

La inserción del nuevo nodo puede provocar la violación de alguna de las reglas. Estas son las posibles ubicaciones de X relativas a P y G.

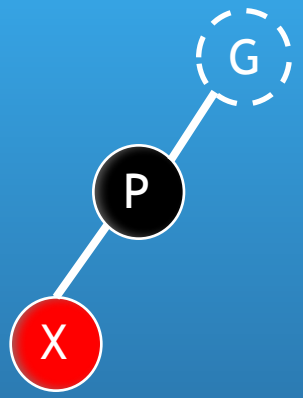


Sin embargo, los nodos pueden ser reordenados utilizando sólo tres métodos (independientemente de los casos mencionados anteriormente).

Insertando un nuevo Nodo

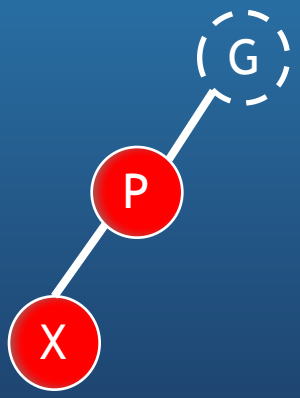
2. Rotaciones después de insertar el Nodo

Las tres posibilidades



CASO 1: P es negro.

- X es rojo, P es negro => no hay conflicto con regla 3
- Tampoco hay aumento de nodos negros.
- PUEDO INSERTAR SIN PROBLEMAS



CASO 2: P es rojo y X es externo. Necesitamos una rotación y algunos cambios de color:

1. Cambie el color de G (el abuelo de X)
2. Cambie el color de P (el padre de X)
3. Rote con G como top y en la dirección que se eleva X (derecha).

Ejemplo en pág. 451 - Figura 9.14

RBTree

applet

CASO 3: P es rojo y X es interno. Necesitamos dos rotaciones y algunos cambios de color.
El truco es hacer una rotación para quedar en la situación del caso anterior.

1. Cambie el color de G (el abuelo de X)
2. Cambie el color de X
3. Rote con P como top y en la dirección que se eleva X.
4. Rotar con G como top y en la dirección que se eleva X.

Ejemplo en pág. 452 - Figura 9.15

Insertando un nuevo Nodo

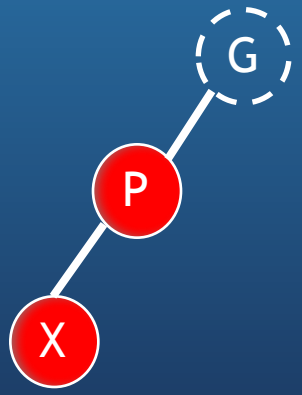
3. Rotaciones en el recorrido hacia abajo

En realidad, esta operación ocurre antes de que el nodo sea insertado.

Los tres casos posibles, son equivalentes a los casos 2 y 3 presentados anteriormente:

- **El nodo que rompe la regla es externo.**
- **El nodo que rompe la regla es interno.**

CASO 2:
P es rojo y X es externo.



RBTree

applet

CASO 3:
P es rojo y X es interno.

Eliminación de Nodos

- Como vimos anteriormente, el eliminar nodos en arboles binarios no es trivial.
- Hacer lo mismo en árboles Rojo-Negro agrega aun más complejidad al proceso, dado que es necesario restaurar la consistencia de los nodos después de la eliminación.
- Solución: Sólo marcar los nodos como eliminados.

Eficiencia de Árboles Rojo-Negro

- Al igual que en los arboles binarios, la búsqueda, inserción y eliminación en un árbol rojo-negro es de orden: $O(\log_2 N)$.
- La única desventaja con respecto a la memoria, se relaciona con el espacio que ocupan la variable booleana que identifica el color del nodo.

Resumen

- Para que la búsqueda en un árbol binario sea rápida, es importante que el árbol permanezca balanceado.
- Ingresando datos ordenados a un árbol binario, provocan que su desbalanceo sea máximo, con tiempo de búsqueda: O(N)
- En el esquema de árbol rojo-negro, cada nodo recibe un color respectivo.
- Un conjunto de reglas, definen formas en las cuales los nodos pueden ser acomodados.
- Estas reglas son utilizadas en la inserción (o eliminación de nodos)
- Un intercambio (flip) de color cambia un nodo negro con dos hijos rojos en un nodo rojo con dos hijos negros.
- En una rotación, se designa a un nodo como el nodo “top”

Resumen

- Una rotación a la derecha, mueve el nodo top en la posición de su hijo derecho, y a su hijo de la izquierda a su posición.
- Una rotación a la izquierda, mueve el nodo top en la posición de su hijo izquierdo, y a su hijo de la derecha a su posición.
- Los intercambios de color y a veces las rotaciones son aplicadas durante el recorrido del árbol (hacia abajo) para encontrar el punto de inserción de un nuevo nodo. Estos cambios permiten conservar las reglas rojo-negro y simplifican el proceso posterior de inserción.
- Después de que el nodo es insertado, se comprueba nuevamente si existen conflictos en las reglas rojo-negro. En su defecto, serán aplicadas algunas rotaciones para obtener un árbol RB correcto.
- Los ajustes anteriormente nombrados generan un árbol balanceado o semi-balanceado.
- El agregar balanceo bajo las reglas rojo-negro sólo agrega una pequeña carga que influye en el rendimiento del árbol, pero evita el caso extrema de bajo rendimiento generado cuando los datos son ingresados al árbol en forma ordenada.

Experimentos

- Haga un diagrama de flujo de todos los movimientos de nodos y colores posibles al insertar un nodo en un árbol rojo-negro, y que se debiera hacer en cada situación para insertar un nuevo nodo.
- Use el applet RBTree para representar todas las situaciones definidas en el experimento 1 y siga las instrucciones para la inserción.
- Haga suficientes inserciones hasta convencerse de que si las reglas 1,2 y 3 son seguidas, la regla 4 se cumple por si sola.