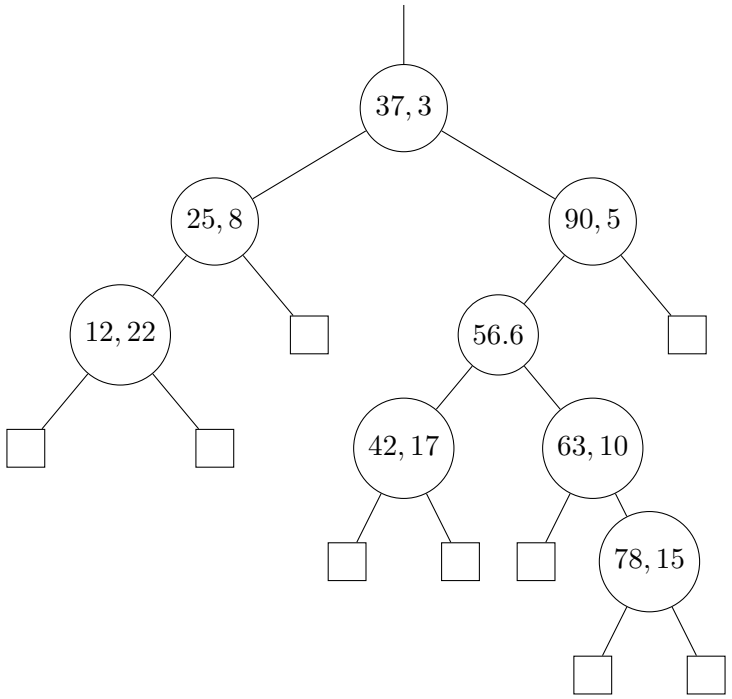
Universidad de Chile
Departamento de Ciencias de la Computación

CC3001 **Algoritmos y Estructuras de Datos** **Otoño 2018**
Nelson Baloian, Jérémy Barbay, Patricio Poblete

Tarea 4
(Fecha de entrega: A definir ulteriormente)

Árboles Cartesianos y Treaps

Un *árbol cartesiano* es un árbol binario en que cada nodo interno almacena un par ordenado (x, y), donde que las coordenadas x forman un árbol de búsqueda binaria (ABB), y las coordenadas y forman un árbol de prioridad (al estilo heap), en que la raíz tiene el mínimo valor de y. Una manera de interpretar la prioridad y es como la hora a la que fue insertada la llave x. El siguiente es un ejemplo:



Supongamos ahora que queremos insertar un nuevo par ordenado $(\bar{x}, \bar{y})$, donde $\bar{y}$ es una prioridad arbitraria, no necesariamente mayor que todas las que están presentes en el árbol, para lo cual usamos el siguiente algoritmo:

a. Primero efectuamos una inserción como en un ABB común y corriente, lo cual deja al nuevo par ordenado como un nodo en el nivel inferior del árbol.

b. Luego, en caso de ser necesario, vamos haciendo rotaciones simples hacia arriba hasta que el nuevo nodo quede bien ubicado de acuerdo a la prioridad.

Parte 1

Suponga que las llaves x son números enteros y que las prioridades y son números reales. Se pide implementar y documentar los métodos

a. `insertar(x,y)`

b. `imprimir()` y

c. `costoPromedio()`

Utilice su programa para procesar las secuencias de inserciones que se le entregarán en archivos que usted deberá leer (con una línea por cada par ordenado). Imprima en cada caso el árbol resultante (en postorden, i.e. en notación polaca, mostrando los nodos externos como "[]") y el costo esperado de una búsqueda exitosa en ese árbol. Por ejemplo, el árbol del ejemplo se imprimiría como:

[][](12,22)[](25,8)[][](42,17)[][][](78,15)(63,10)(56,6)[](90,5)(37,3)

Parte 2

Un *treap* es un árbol cartesiano, en que las prioridades son elegidas al azar en el momento de la inserción. Puede usar $Math.random()$ para generar las prioridades aleatorias. Utilice su implementación de la Parte 1 para repetir el siguiente experimento:

> A partir de un *treap* vacío, insertar las llaves del 1 al n y calcular el costo promedio de búsqueda exitosa en el árbol resultante.

a. Hacer lo anterior para $n = 1024, 2048, \ldots, 65536$, repitiendo 10 veces para cada valor de n.

b. Utilizar los datos resultantes para calcular el costo promedio de búsqueda exitosa para cada n.

c. Graficar esto en función de $\log_2 n$ y estimar el coeficiente que multiplica al logaritmo.

Condiciones de Entrega

- Esta tarea debe ser resuelta en Java.
- Es obligatoria la entrega de un informe junto con su tarea (Ver siguiente sección).
- Esta tarea es de carácter individual, cualquier caso de copia se evaluará con la nota minima.
- No olvide subir a U-cursos todos los archivos necesarios para que su tarea funcione correctamente.
- Debe subir los archivos de código fuente (*.java). Los archivos compilados (*.class) no serán evaluados.
- Cualquier duda respecto a la tarea puede ser consultada usando el foro del curso. No se aceptarán atrasos.

Informe

El informe debe describir el trabajo realizado, la solución implementada, los resultados obtenidos y las conclusiones o interpretaciones de estos. Principalmente debe ser breve, describiendo cada uno de los puntos que a continuación se indican:

- Portada: Indicando número de la tarea, fecha, autor, email, código del curso, etc.
- Introducción: Descripción breve del problema y su solución.
- Análisis del problema: Exponga en detalle el problema, los supuestos que pretende ocupar, casos de borde y brevemente la metodología usada para resolverlo.
- Solución del problema: Indique claramente los pasos que siguió para llegar a la solución del problema. Muestre mediante figuras y ejemplos qué es lo que realiza su código. Evite copiar todo el código fuente en el informe, sin embargo, puede mostrar las partes relevantes de éste.
- Modo de uso: Explicando brevemente cualquier dato necesario para la compilación y ejecución de su programa.
- Resultados: Entregue las salidas que se pide de sus programas., incluyendo gráficos y análisis de los resultados.