



Práctica 2: Divide y vencerás

1. CONSIDERACIONES GENERALES

1.1. ENTREGA DE LA PRÁCTICA

- La entrega de la práctica se realizará en *hendrix-ssh* utilizando el siguiente comando:

```
hendrix:> someter ab_22 practica2.tar
```

- El fichero `practica2.tar` contendrá un directorio denominado `practica2_NIA1_NIA2` (siendo NIA1 y NIA2 los números identificadores de cada estudiante asignados por la Universidad de Zaragoza, y NIA1 será el NIA menor. En el caso de un grupo de práctica formado por un único alumno, el directorio tendrá como nombre `practica2_NIA` (con el identificador de ese alumno)).
- El directorio incluirá los siguientes ficheros de texto:
 - Descripción general del directorio: cómo está organizado, instrucciones de compilación y ejecución, instrucciones para repetir las pruebas, etc. (tiene que llamarse `LEEME`).
 - Listados del código debidamente comentados. Deberán seguir una estructura lógica para poder encontrar y navegar adecuadamente cada una de las partes de la práctica.
 - Un programa para la *shell* denominado `ejecutar.sh` que automatice la compilación y ejecución de los programas entregados con los casos de prueba. Deberá funcionar en *hendrix*.
 - Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
- El directorio incluirá también un informe con la presentación y análisis de resultados (fichero PDF, máximo 3 páginas sin portada). Indicar: nombre, apellidos y NIA de cada miembro del grupo de práctica.
- La fecha límite de entrega para la primera convocatoria** es el día anterior a la sesión de evaluación de las prácticas. Es decir:

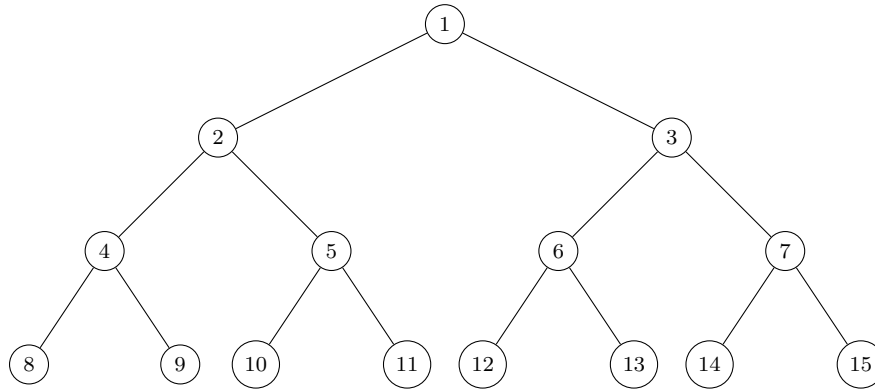
Grupo	Fecha y hora
Jueves A	12/04/2023 8:00AM
Viernes A	13/04/2023 8:00AM
Miércoles A	18/04/2023 8:00AM
Jueves B	19/04/2023 8:00AM
Viernes B	20/04/2023 8:00AM
Miércoles B	25/04/2023 8:00AM

1.2. EVALUACIÓN

- En la calificación se tendrán en cuenta los siguientes aspectos: documentación, diseño e implementación, diseño de casos de prueba, análisis de las pruebas realizadas y facilidad para la repetición de las pruebas por los profesores.
- Se aplicarán las reglas de tratamiento de casos de plagio explicadas en la presentación de la asignatura.

2. ENUNCIADO

Como si fuera una máquina de pinball se dejan caer una cantidad de n bolas, una por una, desde la raíz de una estructura de árbol binario completo. Todos los nodos del árbol están numerados de forma secuencial, comenzando en 1 con la raíz (nivel 1), y luego los hijos de nivel 2, y así sucesivamente. Los nodos en cualquier nivel están numerados de izquierda a derecha. Cada vez que una bola se deja caer, visita primero un nodo no terminal del árbol, continúa moviéndose hacia abajo, ya sea siguiendo el camino del sub-árbol izquierdo o el camino del sub-árbol derecho, hasta que llega en uno de los nodos hoja del árbol. Para determinar la dirección de movimiento de una bola, cada nodo no terminal tiene asociado un valor booleano (*flag*): falso o verdadero. Al comienzo del juego, todos los *flag* tienen valor falso. Al visitar un nodo no terminal, si el valor actual del *flag* en este nodo es falso, la bola primero cambiará el valor del nodo, es decir, de falso a verdadero, y luego seguirá el sub-árbol izquierdo de este nodo para seguir moviéndose hacia abajo. De lo contrario, también cambiará el valor del *flag*, es decir, de verdadero a falso, pero seguirá el sub-árbol derecho de este nodo para seguir moviéndose hacia abajo. Por ejemplo, la figura a continuación representa un árbol binario completo de nivel 4.



Puesto que todos los *flag* están configurados inicialmente como falsos, la primera bola que se deja caer cambiará los valores de los *flag* en los nodos 1, 2 y 4 antes de que finalmente se detenga en la posición 8. La segunda bola que se suelta cambiará los valores de los *flag* en los nodos 1, 3 y 6, y se detendrá en la posición 12. La tercera bola que se deja caer cambiará los valores de los *flag* en los nodos 1, 2 y 5 antes de que se detenga en posición 10.

TAREA 1. DISEÑO Diseña un algoritmo *divide y vencerás* que tome en entrada: 1) la profundidad máxima P del árbol y 2) la i -ésima bola, y determine la posición de parada de la bola. Puedes asumir que el valor i es menor o igual al número total de hojas del árbol.

TAREA 2. IMPLEMENTACIÓN Desarrolla un programa que genere el árbol con profundidad P y calcule la posición de las primeras n bolas conforme al algoritmo de solución propuesto. El programa tendrá que verificar también la coherencia de los valores de P y n (el decir, el número

n tiene que ser menor o igual al número total de hojas del árbol). En particular, la forma de ejecutar el programa será el siguiente:

```
hendrix:> pinball pruebas.txt resultados.txt
```

donde:

- **pruebas.txt** es un fichero de texto que incluye diferentes configuraciones de los parámetros del problema (donde los parámetros son la profundidad P del árbol y el número n de bolas que se quiere dejar caer), y
- **resultados.txt** es un fichero de texto que guarda los resultados. En particular, para cada configuración se guardarán: 1) las posiciones de las bolas y 2) los tiempos de ejecución del programa.

TAREA 3. EXPERIMENTACIÓN Verifica la corrección del programa y analiza la eficiencia (tiempos de ejecución) de los algoritmos implementados (es decir, la generación del árbol y el cálculo de las posiciones de las bolas) con casos de prueba.