

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA
Segundo Examen
(Segundo Semestre de 2013)

Horario 0581: prof. Andrés Melgar
Horario 0582: prof. Fernando Alva

Duración: 3 horas

Nota:

- No se permite el uso de material de consulta.
- No se ofrecerá asesoría en la parte teórica.
- **Debe utilizar comentarios para explicar la lógica seguida en los programas elaborados, así como nombres de variables apropiados.**
- La presentación, la ortografía y la gramática influirán en la calificación.

Puntaje total: 20 puntos

Cuestionario:

PARTE TEÓRICA

Responda las siguientes preguntas según los conceptos vistos en clase:

Pregunta 1 (1 punto) ¿Es correcto afirmar que el uso de árboles binarios mejora las performance de las búsquedas en un conjunto de datos? Justifique su respuesta.

Pregunta 2 (1 punto) ¿En qué situaciones conviene representar grafos usando lista de adyacencia?

Pregunta 3 (1 punto) ¿Qué diferencia a un tipo abstracto de datos de una estructura de datos?

Pregunta 4 (1 punto) A nivel de implementación: ¿Cuál es la diferencia entre la búsqueda en amplitud y la búsqueda en profundidad?

Pregunta 5 (1 punto) ¿En qué se diferencia un árbol de un grafo? ¿Se puede decir que un árbol es un caso particular de grafo? Justifique su respuesta.

PARTE PRÁCTICA

Pregunta 6 (5 puntos) Hacer una función que recibiendo como único parámetro un puntero a un árbol binario y sin usar otras funciones definidas por el usuario, permita calcular la suma del valor de la hoja de menor nivel y el valor de la hoja de mayor nivel del árbol. Si existe más de una hoja en el mismo nivel, se deberá tomar la de más a la izquierda.

Pregunta 7 (6 puntos) UVa 10801 - Lift Hopping (Traducción Libre) Un rascacielos que no tiene más de 100 pisos (enumerados de 0 a 99) posee n ($1 \leq n \leq 5$) ascensores que viajan de arriba a abajo a (posiblemente) diferentes velocidades. Para cada i en $\{1, 2, \dots, n\}$, el ascensor número i demora T_i ($1 \leq T_i \leq 100$) segundos en viajar entre dos pisos adyacentes cualesquiera (yendo hacia arriba o hacia

abajo). Los ascensores no necesariamente se detienen en todos los pisos. Lo que es peor, no todos los pisos son necesariamente accesibles por un ascensor.

Usted se encuentra en el piso 0 del rascacielos y le gustaría llegar al piso k tan rápido como sea posible. Asuma que no necesita esperar para abordar el primer ascensor y, por simplicidad, la acción de cambiar de ascensores en algún piso siempre toma exactamente un minuto. Claro, ambos asesores deben parar en dicho piso. Está prohibido usar las escaleras. Nadie más está en el ascensor con usted, así que no necesita detenerse si no lo desea. Calcule el número mínimo de segundos requeridos para llegar desde el piso 0 hasta el piso k (pasar por el piso k mientras está dentro de un ascensor que no para en dicho piso no cuenta como “llegar al piso k ”).

Entrada: La entrada consistirá de diferentes casos de prueba. Cada uno comenzará como dos números, n y k , en una línea. La siguiente línea contendrá los números T_1, T_2, \dots, T_n . Finalmente, las siguientes n líneas contendrán listas ordenadas de números enteros: la primera línea listará los pisos que puede visitar el ascensor número 1, la siguiente listará los pisos que puede visitar el ascensor número 2, etc.

Salida: Para cada caso de prueba, debe imprimir un número en una única línea que indique el número mínimo de segundos requeridos para llegar al piso k desde el piso 0 del rascacielos. Si no es posible hacerlo, imprima “IMPOSIBLE”.

Ejemplo de Entrada:

```
2 30
10 5
0 1 3 5 7 9 11 13 15 20 99
4 13 15 19 20 25 30
2 30
10 1
0 5 10 12 14 20 25 30
2 4 6 8 10 12 14 22 25 28 29
3 50
10 50 100
0 10 30 40
0 20 30
0 20 50
1 1
2
0 2 4 6 8 10
```

Ejemplo de Salida:

```
275
285
3920
IMPOSIBLE
```

Explicación de los Ejemplos:

- En el primer caso de prueba, tome el ascensor 1 hasta el piso 12 (130 segundos), espere 60 segundos para cambiar al ascensor 2 y viaje hasta el piso 30 (85 segundos) para un total de 275 segundos.
- En el segundo caso de prueba, tome el ascensor 1 hasta el piso 10, cambie al ascensor 2 y viaje hasta el piso 25. Allí, cambie al ascensor 1 y viaje hasta el piso 30. El tiempo total es $10 \cdot 10 + 60 + 15 \cdot 1 + 60 + 5 \cdot 10 = 285$ segundos.
- En el tercer caso de prueba, tome el ascensor 1 hasta el piso 30, luego el ascensor 2 hasta el piso 20 y luego el ascensor 3 hasta el piso 50.

- En el último caso de prueba, el único ascensor disponible no llega al piso 1.

Implemente un programa en ANSI C que resuelva el problema planteado. El tiempo límite de ejecución del programa es de 1 segundo.

PARTE ELECTIVA

Responda **UNA** de las siguientes preguntas:

Pregunta 8 (4 puntos) Implemente una función en ANSI C que permita encontrar el k -ésimo elemento desde el final de una lista simplemente enlazada. No se conoce el tamaño de la lista. El algoritmo debe demorar $O(n)$ y ocupar $O(1)$ de espacio adicional al ocupado por la lista. Debe indicar la estructura de datos que está usando para implementar la lista y sus nodos. No implemente operaciones de manejo de listas que no intervengan directamente en la implementación de su función.

Pregunta 9 (4 puntos) ¿Cómo diseñaría una Pila que, además de *push* (apilar) y *pop* (desapilar), también posea una operación **min** que retorne el menor elemento en la Pila? *Push*, *pop* y *min* deben operar en tiempo $O(1)$. Explique la estrategia a seguir e implemente su solución en ANSI C.

Profesores del curso: Andrés Melgar
 Fernando Alva

Pando, 07 de diciembre de 2013