



UNIVERSIDAD RICARDO PALMA
EXAMEN PARCIAL DE ESTRUCTURAS DE DATOS Y ALGORÍTMICA
ESCUELA DE INGENIERÍA INFORMÁTICA
Semestre 2012-1 Fecha: Viernes 11 de Mayo del 2012

- El examen es sin material de consulta
- Todos los códigos o programas deberán ser escritos en Java.
- Tiempo de realización: 120 minutos.
- La entrega del examen calificado será en horas de clase en la semana siguiente después del parcial.
- **Esta hoja deberá ser entregada al finalizar el examen, junto con la hoja de respuestas.**
- El puntaje de una pregunta será cero (0), si no la responde o no está de acuerdo a lo que exactamente se pide.

APELLIDOS Y NOMBRE:

CÓDIGO:

1. (2 pts.) Explique todos los pasos del proceso de solución de problemas.
2. (2 pts.) Con respecto a las listas de acceso aleatorio y de acceso secuencial, explique cuándo es preferible utilizar una en vez de la otra. Explique ambos casos.
3. (6 pts.) Escriba la función "int buscaSubSecuencia(int a[], int na, int b[], int nb)", en donde *na* y *nb* son la cantidad de enteros válidos que tienen los arreglos *a* y *b*, respectivamente, desde el índice cero (0). La función debe retornar el índice en *a* en donde comience la primera ocurrencia de la secuencia de enteros de *b*; debe retornar -1 si *a* no contiene la secuencia *b*.
NO DEBE USAR SENTENCIAS DE BUCLES: while, for, do...while,...

Sugerencia:

```
private static boolean contieneDesdeIAux(int[] a, int na, int[] b, int nb, int i, int k)
{ //auxiliary de contieneDesdeI
}
public static boolean contieneDesdeI(int[] a, int na, int[] b, int nb, int i) {
//retorna VERDADERO si b está contenido en a, desde a[i]
}
private static int buscaSubSecuenciaAux(int[] a, int na, int[] b, int nb, int i) {
//busca una subsecuencia de b en a, desde el índice i
}
public static int buscaSubSecuencia(int[] a, int na, int[] b, int nb) {
//busca una subsecuencia de b en a
}
}
```

4. Escriba el código del operador `remove` (suprimir) correspondiente a una implementación de lista de acceso **secuencial CIRCULAR**. Responda **a)** o **b)**.
 - a) (4 pts.) `remove` y el método que se invoca desde éste: `nodo`.
 - b) (2 pts.) Únicamente el método `remove`.
 - c) (0 pts.) Si usted **NO** considera una lista de **secuencial CIRCULAR** o si contesta más de una.
5. Escriba el código del operador `remove` (suprimir) correspondiente a una implementación de **lista de acceso aleatorio**, pero con la siguiente restricción:

"Se podrá eliminar siempre y cuando el elemento sea mayor o igual al promedio de todos los elementos anteriores; es decir, al promedio de todos los elementos cuyo índice es menor que el indicado como parámetro".

`remove` sólo debe tener el único parámetro: el índice, `index`. Si el índice es el cero (0), el elemento puede eliminarse sin ninguna restricción. Responda **a)** o **b)**.

- a) (6 pts.) Si usted responde tal como indica el enunciado de la pregunta.
- b) (1 pts.) Si usted escribe el operador `remove` sin tomar en cuenta la restricción expresada en el enunciado.
- c) (0 pts.) Si usted **NO** considera una **lista de acceso aleatorio** o si contesta más de una.

Los profesores del curso