

PROGRAMACIÓN II

Profesor: David E. Losada

Examen
Julio 2010

Fecha: 12/7/2010.

Nombre y apellidos: _____

1. Test (3 puntos. Cada respuesta correcta suma 0.2 puntos. Cada respuesta incorrecta resta 0.1 puntos)

1. Un Tipo Abstracto de Dato (TAD) permite
 - a. ocultar los detalles de implementación de un nuevo de tipo de datos
 - b. hacer explícitos los detalles de programación asociados a un nuevo de tipo de datos
 - c. abstraerse de las operaciones que se harán sobre el tipo (esto es, no se concretan)
 - d. ninguna de las anteriores
2. En verificación de software una falla es
 - a. un estado intermedio incorrecto en que entra un programa al ejecutarse
 - b. un síntoma de que existe un error
 - c. un caso de prueba
 - d. ninguna de las anteriores
3. La verificación formal de programas está basada
 - a. en probar los programas con casos de prueba completos
 - b. en conseguir variedad de usuarios que prueben los programas
 - c. en especificaciones matemáticas sobre lo que se espera de las operaciones del programa
 - d. ninguna de las anteriores
4. La técnica de ramificación y poda...
 - a. es una variante de la fuerza bruta
 - b. realiza un recorrido ciego del espacio de búsqueda
 - c. realiza un recorrido informado del espacio de búsqueda
 - d. ninguna de las anteriores
5. Los algoritmos de vuelta atrás o backtracking...
 - a. son similares en filosofía a los algoritmos voraces (no reconsidera decisiones tomadas previamente)
 - b. realizan una búsqueda dentro de un espacio de búsqueda que se suele representar de forma arbórea
 - c. nunca realizan un recorrido en profundidad del espacio de búsqueda
 - d. ninguna de las anteriores
6. Un algoritmo voraz
 - a. divide el problema en subproblemas de modo recursivo
 - b. suele ser más ineficiente temporalmente que uno de backtracking
 - c. es un caso especial de ramificación y poda
 - d. ninguna de las anteriores
7. La búsqueda binaria

- a. es un ejemplo de búsqueda secuencial
 - b. es un ejemplo de búsqueda lineal
 - c. es un ejemplo de fuerza bruta
 - d. ninguna de las anteriores
8. Se dispone de un algoritmo superlineal, un sublineal y un cuadrático que resuelven el mismo problema. en general, ¿cual elegimos?
- a. sublineal
 - b. superlineal
 - c. cuadrático
 - d. depende de las constantes multiplicativas
9. El orden inferior de complejidad (omega) nos da una idea de
- a. el comportamiento del algoritmo en el mejor de los casos
 - b. el comportamiento del algoritmo en el caso promedio
 - c. el comportamiento del algoritmo en el caso de tallas pequeñas
 - d. ninguna de las anteriores
10. Si tenemos un algoritmo lineal y otro logarítmico que resuelven el mismo problema
- a. el logarítmico siempre es mejor
 - b. el lineal siempre es mejor
 - c. para tallas grandes el lineal pudiera ser mejor
 - d. ninguna de las anteriores
11. ¿por qué se suele realizar en análisis de algoritmos centrándonos en el comportamiento para problemas de talla grande?
- a. el análisis de algoritmos no se realiza de ese modo
 - b. porque para problemas pequeños la elección del algoritmo es menos importante porque los problemas pequeños se resuelven rápidamente en todo caso
 - c. porque así aseguramos el comportamiento del algoritmo en el mejor de los casos
 - d. ninguna de las anteriores
12. La búsqueda con centinela respecto a la búsqueda sin centinela
- a. mejora el orden de complejidad
 - b. empeora el orden de complejidad
 - c. el orden de complejidad es el mismo pero se realizan menos pasos
 - d. ninguna de las anteriores
13. El algoritmo quicksort de ordenación
- a. es un ejemplo de algoritmo voraz
 - b. es un ejemplo de divide y vencerás
 - c. es un ejemplo de vuelta atrás
 - d. ninguna de las anteriores
14. Un TAD pila se implementa de dos maneras alternativas (A y B). La alternativa A es una lista enlazada de elementos con un puntero a la cima de la pila. La alternativa B es mediante un bloque de memoria contigua.

¿cual de las siguientes afirmaciones es cierta?

- a. un problema con la alternativa B es que tengo que decidir cuantos elementos reservo a priori

- b. un problema con la alternativa B es que necesito un puntero adicional para cada valor almacenado en la pila
- c. un problema con la alternativa A es decidir cuantos elementos reservo memoria en el bloque
- d. ninguna de las anteriores

15. Un TAD lista se implementa de dos maneras alternativas (A y B). La alternativa A es un puntero a un bloque contiguo de elementos. La alternativa B es mediante una lista doblemente enlazada con un puntero al primer elemento de la lista y doble enlace entre cada elemento de la lista (hacia delante y hacia atrás).

¿cual de las siguientes afirmaciones es cierta?

- a. la alternativa A me facilita la gestión de los accesos, tanto para inserciones como para eliminaciones de elementos de la lista
- b. para obtener el n-ésimo elemento la alternativa A es más eficaz pues los elementos están dispuestos de manera contigua en memoria
- c. un problema con la alternativa A es que utiliza más memoria que la B.
- d. ninguna de las anteriores

2. (1.5 punto) Construye la especificación formal del TAD Natural con los constructores Cero y Sucesor y las operaciones EsCero, Igual, Suma, Antecesor y Diferencia.

3. (2.5 puntos) Una bibliotecaria dispone de tres pilas de libros y quiere construir una lista con todos los libros distintos que hay en las tres pilas. Cada libro se identifica en la pila por su isbn (esto es, el tipo elemento es un long que almacena el isbn del libro). Construye una función que tome esas tres pilas, construya una lista con los libros distintos que hay (sin repeticiones) y, finalmente, deje las tres pilas como estaban. Utiliza para ello las operaciones conocidas en los TADs Pila (PilaVacía, EsVacía, Cima, Push y Pop) y Lista (crea, fin, primero, siguiente, anterior, esVacía, recupera, longitud, inserta, suprime, modifica) realice esta operación.

```
// rellena los argumentos y tipo que recibe la función
void crea_Lista_dadas_3_Pilas(
{
// rellena las instrucciones necesarias

.....

}
```

4. (3 puntos) Dado los siguientes programas, determina el orden de complejidad de los mismos siendo la talla del problema el número N

Programa A:

.... int main() { k = f1(N)+f2(N); }	float f1(int num) { int aux, aux2, aux3; float rdo=0; for (aux=0;aux<num;aux++) for (aux2=0;aux2<num;aux2++) for (aux3=0;aux3<num;aux3++) rdo += aux*aux2*aux3; return rdo; }	float f2(int num) { int aux,aux2; float rdo=0; for (aux=0;aux<num;aux++) for (aux2=0;aux2<num;aux2++) rdo += aux+aux2; return rdo; }
--	--	--

Programa B:

.... int main() { k = f1(N); }	float f1(int num) { float rdo; if (num < 5) return (num*1.1) rdo= 2*f1(num-1)*f1(num-2); return(rdo); }
--	--

Programa C:

.... int main() { k = f1(N)+f2(N); }	float f1(int num) { int aux, aux2; float rdo=0; for (aux=0;aux<num;aux++) for (aux2=0;aux2<num;aux2++) rdo += aux*aux2; return rdo; }	float f2(int num) { int aux; float rdo=0; for (aux=0;aux<num;aux++) rdo += aux; return rdo; }
--	---	--