Algoritmos y Estructuras de Datos II – 21 de Febrero de 2022

Examen Final Teórico-Práctico

Alumno: ..... Email: .....

Siempre se debe explicar la solución. Una respuesta correcta no es suficiente si no viene acompañada de una justificación lo más clara y completa posible. Los algoritmos no deben escribirse utilizando código c o de bajo nivel, sino el código de la materia y evitando la utilización innecesaria de punteros. La no observación de estas recomendaciones resta puntaje.

1. (Algoritmos voraces) Estás en época de exámenes y tenés n materias cursadas, no correlativas entre sí, que podrías rendir. Cada materia tiene un día de examen: $d_1, \ldots, d_n$, y una cantidad de días previos **consecutivos** al examen que vos necesitás dedicar exclusivamente a su estudio: $c_1, \ldots, c_n$. También asumimos que el día que rendís un examen se dedica solamente a eso, no podés estudiar otra materia. Así por ejemplo si la materia "Bases de Datos" se rinde el día 10, y necesita 2 días de estudio, para poder rendirla tenés que dedicar el día 8 y 9 exclusivamente a la misma, y en el día 11 ya podrías empezar a estudiar otra materia. Se supone que solo estudiás la materia que estás por rendir, por más que te sobren días no comenzás a estudiar la siguiente para no confundir los temas.

   Todos los d_i y los c_i son números naturales, inicialmente estamos al comienzo del día 1.

   Se debe obtener la mayor cantidad de materias que podés rendir.

   Se pide lo siguiente:

   (a) Indicar de manera simple y concreta, cuál es el criterio de selección voraz para construir la solución?

   (b) Indicar qué estructuras de datos utilizarás para resolver el problema.

   (c) Explicar en palabras cómo resolverá el problema el algoritmo.

   (d) Implementar el algoritmo en el lenguaje de la materia de manera precisa.

2. (Backtracking)

   El presidente de tu país te acaba de elegir como asesor para tomar una serie de medidas de producción que mejoren la situación económica. En el análisis preliminar se proponen n medidas, donde cada medida $i \in \{1, .., n\}$ producirá una mejora económica de m_i puntos, con $m_i > 0$. También se analizó para cada una el nivel de daño ecológico d_i que producirá, donde $d_i > 0$. El *puntaje* que tendrá cada medida i está dado por la relación m_i/d_i.

   Se debe determinar cuál es el máximo puntaje obtenible eligiendo K medidas, con $K < n$, de manera tal que la suma total del daño ecológico no sea mayor a C.

   Se pide lo siguiente:

   (a) Especificá precisamente qué calcula la función recursiva que resolverá el problema, indicando qué argumentos toma y la utilidad de cada uno.

   (b) Da la llamada o la expresión principal que resuelve el problema.

   (c) Definí la función en notación matemática.

3. (Programación Dinámica) Implementá un algoritmo que utilice Programación Dinámica para resolver el problema del inciso anterior.

   - ¿Qué dimensiones tiene la tabla que el algoritmo debe llenar?
   - ¿En qué orden se llena la misma?
   - ¿Se podría llenar de otra forma? En caso afirmativo indique cuál.

4. Para cada uno de los siguientes algoritmos determinar **por separado** cada uno de los siguientes incisos.

   (a) ¿Qué hace? ¿Cuáles son las precondiciones necesarias para haga eso?

   (b) ¿Cómo lo hace?

   (c) El orden del algoritmo, analizando los distintos casos posibles.

   (d) Proponer nombres más adecuados para los identificadores (de variables, funciones y procedimientos).

```
fun s(p: array[1..n] of nat, v,w: nat) ret y: nat
    y:= v
    for i := v+1 to w do
        if p[i] < p[y] then y:= i fi
    od
end fun

fun t(p: array[1..n] of nat, v,w: nat) ret y: nat
    y:= v
    for i := v+1 to w do
        if p[y] < p[i] then y:= i fi
    od
end fun
```

```
proc r(p: array[1..n] of nat)
    for i := 1 to n div 2 do
        swap(p, i, s(i, n-i+1));
        swap(p, n-i+1, t(i+1, n-i+1));
    od
end fun
```

5. (TADs) Te contratan para diseñar un software de billetera virtual, la cual puede almacenar saldo en tres monedas distintas: Peso, Real y Dólar. Cada usuario puede recibir pagos en cualquiera de las tres monedas, y puede realizar pagos en alguna de las tres monedas, siempre y cuando tenga el saldo suficiente.

   Se pide:

   (a) **Especificar** el TAD **Wallet** mediante un constructor que cree la wallet con saldo 0 en las tres monedas. Además debe proveer operaciones para averiguar cuál es el saldo en cada moneda, para recibir pagos en cada una de las monedas y para realizar pagos con cada una de ellas también. Las operaciones de recibir y realizar pagos deben especificarse como **procedimientos** que modifiquen una Wallet.

   La especificación debe realizarse utilizando el lenguaje visto en la materia, indicando el tipo de cada operación, la precondición en caso que tenga y un comentario describiendo qué hace.

   (b) **Implementar** el tad utilizando una tupla con 3 números. Se debe utilizar precisamente el lenguaje de la materia.

   (c) Utilizando el tipo **abstracto**, se debe implementar una operación que, dado un número racional indicando la relación entre dólar y peso, modifique una wallet convirtiendo todos los dólares que tenga en el saldo, a su correspondiente en pesos. Se debe utilizar precisamente el lenguaje de la materia.

6. (Para alumnos libres) Algunos problemas pueden resolverse utilizando backtracking, y también utilizando programación dinámica. En general, ¿cuáles son las ventajas y desventajas de una de estas técnicas en comparación con la otra desde el punto de vista de la eficiencia?