Apellido y Nombre:
e-mail:

escrito	lab	nota

1	2	3	4	5	L

Algoritmos y Estructuras de Datos II **Examen Final** 23/7/2008

Importante: Escribir nombre y apellido en todas las hojas, inclusive ésta que también debe ser entregada. Resolver cada ejercicio en hoja aparte.

1. (2,5pts) Especificar el TAD ReservaAulas de modo de que refleje las reservas que se van haciendo de las aulas. El tipo tendrá sólo dos constructores, uno que representa la situación inicial (en la que no hay ningún aula reservada) y otro que permite justamente agregar una reserva: un docente reserva un aula a partir de una cierta hora y por un número determinado de horas. Por simplicidad representaremos las horas con números naturales:
   TAD ReservaAulas
   constructores
   nohayresevas :: ReservaAulas
   agregarreserva :: Docente × Aula × Nat × Nat × ReservaAulas → ReservaAulas
   Con esta notación,

   agregarreserva(daniel, aulaA, 9, 4, agregarreserva(juan, aulaB, 11, 3, nohayreservas))

   significa que daniel reservó el aula A de 9 a 13 (es decir, por 4 horas a partir de las 9) y juan el aula B de 11 a 14 (es decir, por 3 horas a partir de las 11). No hay que preocuparse por números naturales mayores que 24 ya que la hora 30 es la hora 6 del día siguiente, la hora 245 es la hora 5 dentro de 10 días, etc. De la misma manera, está permitido reservar un aula por las horas que sea.
   a) Se pide dar ecuaciones que expresen
      1) que agregar una reserva por 0 horas no tiene ningún efecto,
      2) que el orden en que se han agregado reservas de aulas diferentes no tiene importancia
      3) que agregar una reserva que entra en conflicto con otra ya realizada no tiene ningún efecto. Por ejemplo, si en vez de aulaB dijera aulaA en el ejemplo de arriba, daniel no podría haber reservado el aula y lo expresado sería igual a agregarreserva(juan, aulaA, 11, 3, nohayreservas)
      4) que si un docente reserva un aula en forma consecutiva (por ejemplo, de 9 a 11y de 11 a 16 (o de 10 a 16), es lo mismo que hacer una sola reserva de 9 a 16. Lo mismo si el docente reserva de 9 a 18 y luego de 13 a 16, la segunda reserva no debería tener efecto.
   b) Agregar una operación que permita determinar si un aula está libre por un número de horas m a partir de la hora n. Dar la signatura (o tipo) y sus ecuaciones.
   c) (alumnos libres) Agregar una operación que permita determinar si un docente ha reservado algún aula por un número de horas m a partir de la hora n. Agregar otra que permita determinar si un mismo docente ha reservado dos aulas simultáneamente. Dar las signaturas y las ecuaciones.

   Las ecuaciones pueden requerir condiciones en términos de $<$, $=$, $\leq$, etc y operaciones como la suma o resta. Asuma que están definidas.

2. (2pts) Resolver el problema de la sociedad rural: un latifundista tiene S toneladas de soja y quiere almarcenarlas hasta que le aseguremos máxima rentabilidad. Existen n silos con capacidad para $t_1, \ldots, t_n$ toneladas, y el alquiler de cada uno de ellos por el tiempo estimado cuesta $p_1, \ldots, p_n$. Dar un algoritmo que determina el mínimo costo

posible que le será necesario pagar en concepto de alquiler para almacenar toda la soja. Justificar. ¿Qué técnica utilizó?

3. (2pts) Se debe calcular el número de veces que el procedimiento p escribe la palabra "hola" en función de la entrada n. Para ello
   a) Definir la recurrencia correspondiente.
   b) Resolverla.

   Obs: recuerde que usted mismo puede comprobar que la recurrencia esté correctamente resuelta. El argumento de p puede ser cualquier número natural $n \geq 0$.

```
proc p(in n:nat)
    if n = 1 then escribir("hola") fi
    if n ≥ 2 then p(n-1)
                  p(n-2)
    fi
end
```

4. (2pts) Dado un arreglo a de reales entre x e y, se pide dar una variante del algoritmo de ordenación rápida que en lugar de utilizar un valor del arreglo a como pivote, utiliza el valor $(x+y)/2$ (que puede o no estar en el arreglo). Así, qsort debe recibir como parámetros, el arreglo a que se está ordenando, las posiciones izq y der y los valores x e y que, se asume, cumplen con la condición

$$\forall k \in \{izq, \ldots, der\}.a[k] > x \wedge a[k] \leq y$$

   Observar que al usar $(x+y)/2$ como pivote, al no haber garantía de que haya un elemento en a con ese valor, sólo se logra partir $a[izq, der]$ en dos: $a[izq, med]$ y $a[med+1, der]$. A diferencia del algoritmo dado en clase en que además de partir en dos ($a[izq, piv-1]$ y $a[piv+1, der]$) quedaba el pivote $a[piv]$ fuera de esa partición. Justificar.

5. (1,5pts) Se tiene un grafo $G = (N, A)$ donde N es el conjunto de nodos y A el de aristas dirigidas. La función $o : A \to N$ devuelve el origen de cada arista. La función $d : A \to N$ devuelve el destino de cada arista. Finalmente, cada arista tiene asociado un premio. La función $p : A \to$ **nat** devuelve el importe del premio correspondiente a cada arista. Dados $x, y \in N$ el siguiente algoritmo utiliza backtracking para encontrar el máximo premio acumulable al ir de x a y por caminos que no tienen ciclos.

```
fun max_premio(x : N, y : N) ret n: nat
                {calcula el máximo premio acumulable por caminos sin ciclos de x a y}
    n:= backtrack(x,y,{x},0)
end
fun backtrack(x' : N, y : N, V :cjto de N, P :nat) ret n: nat
                {x' es el último nodo visitado en el camino que se está construyendo}
                          {V es el conjunto de los nodos visitados en ese camino}
                              {P es el premio acumulado a lo largo de ese camino}
    if x' = y → n:= P
       x' ≠ y → n:= 0
                for a ∈ A do
                    if o(a) = x' ∧ d(a) ∉ V →
                         n:= max(n,backtrack(d(a),y,V ∪ {d(a)},P + p(a))) fi
                od
    fi
end
```

Se pide explicar el algoritmo lo más detalladamente posible. Justificar claramente cada asignación a n y los valores de cada uno de los parámetros en cada llamada a la función backtrack.

Explicar si el algoritmo siempre termina o no y por qué.

Explicar qué ocurre con grafos no conexos. Y qué ocurre con grafos con ciclos.