Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos. Recuperatorio. [30 de Junio de 2005]

[Ej. 1] [clases (20 pts)]

a) [**parcial-1 lista (20 pts)**] Escribir los siguientes métodos del **TAD lista**: `insert(p,x)`, `erase(p)`, `begin()`.

b) [**parcial-2 arbol-bin (20 pts)**] Escribir los siguientes métodos del **TAD árbol binario**: `find(x,p)`, `insert(p,x)`, `clear()`.

c) [**parcial-3 set (20 pts)**] Escribir los siguientes métodos del **TAD conjunto** por listas ordenadas: `insert(x)`, `find(x)`, `clear()`.

[Ej. 2] [programacion (50 pts)]

a) [**map-pre-post (30 pts)**] Escribir una función `void map_pre_post(tree<int> &T,list<int> &L, int (*fpre)(int),int (*fpost)(int))` que lista los valores nodales del árbol ordenado orientado `T` en una mezcla de orden previo y posterior, de acuerdo a la siguiente definición

$$\text{mpp}(\Lambda, f, g) = \text{lista vacía}$$
$$\text{mpp}(n, f, g) = f(n), \text{mpp}(n_1, f, g), ..., \text{mpp}(n_m, f, g), g(n)$$

donde $n_1 ... n_m$ son los hijos del nodo n. Por ejemplo, si `T=(1 3 (5 6 7 8))`, $f(x) = x$ y $g(x) = x + 1000$, entonces `map_pre_post(T,L,f,g)` debe dar `L=(1,3,1003,5,6,1006,7,1007,8,1008,1005,1001)`.

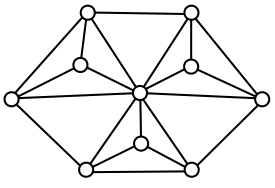
b) [**purge (10 pts)**] Escribir una función `void purge(list<int> & L)` que purga elementos repetidos de una lista usando un conjunto como estructura auxiliar y con una implementación tal que sea $O(n \log n)$.

c) [**cum-sum-pila (10 pts)**] Escribir una función `void cum_sum_pila(stack<int>&S)` que modifica a la pila `S` dejando la suma acumulada de sus elementos, es decir, si los elementos de `S` antes de llamar a `cum_sum_pila (S)` son $S = (a_0, a_1, \ldots, a_{n-1})$, entonces después de llamar a `cum_sum_pila (S)` debe quedar $S = (a_0, a_0 + a_1, \ldots, a_0 + a_1 + \ldots + a_{n-1})$. Por ejemplo, si `S = (1,3,2,4,2)` entonces después de hacer `cum_sum_pila (S)` debe quedar `S = (1,4,6,10,12)`. Restricciones: (i) usar una pila auxiliar; (ii) NO usar más estructuras auxiliares que la indicada ni otros algoritmos de STL; y (iii) el algoritmo debe ser $O(n)$.

[Ej. 3] [operativos (20 pts)]

- [**parcial-1 color-grafo (10 pts)**]

  Colorear el grafo de la figura usando un algoritmo heurístico ávido para obtener el mínimo número de colores posible. ¿La coloración obtenida es óptima? Justifique.



- **[parcial-1 t-exec (10 pts)]**
  Dadas las funciones
  - $T_1(n) = 5n^3 + 2n! + \log n$,
  - $T_2(n) = 2^{15} + 2 \cdot 5^n + 3 \cdot n^2$,
  - $T_3(n) = 6! + n^2 + n^{1.7}$,
  - $T_4(n) = 1.3 \cdot 2^3 + 20n + \log_2 10$.

  ordenarlas de menor a mayor.

  $T_{\square} < T_{\square} < T_{\square} < T_{\square}$
- **[parcial-2 rec-arbol (8 pts)]** Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
  - `ORD_PRE` $=\{A, Z, W, D, E, X, Y, P, Q, R, S\}$,
  - `ORD_POST` $=\{E, D, W, X, Z, P, S, R, Q, Y, A\}$.
- **[parcial-2 huffman (8 pts)]** Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario y encodar la palabra `CONEXA`
  $P(C) = 0.2, P(O) = 0.1, P(N) = 0.1, P(A) = 0.05, P(L) = 0.05, P(X) = 0.1, P(E) = 0.1, P(Q) = 0.3$
  Calcular la longitud promedio del código obtenido.
- **[parcial-2 part-arbol (4 pts)]** Particione el árbol AOO `(z (q w y l) (r t u))` con respecto al nodo `q`, es decir indique cuales son sus antecesores y descendientes propios, derecha e izquierda.
- **[parcial-3 heap-sort (5 pts)]** Dados los enteros $\{21, 12, 9, 3, 5, 10, 7, 6\}$ ordenarlos por el método de *"montículos"* (*"heap-sort"*). Mostrar el montículo (minimal) antes y después de **cada** inserción/supresión.
- **[parcial-3 quick-sort (5 pts)]** Dados los enteros $\{17, 19, 9, 6, 2, 3, 13, 18, 9, 0\}$ ordenarlos por el método de *"clasficación rápida"* (*"quick-sort"*). En cada iteración indicar el pivote y mostrar el resultado de la partición.
- **[parcial-3 abb (5 pts)]** Dados los enteros $\{10, 12, 15, 11, 7, 6, 4, 5, 8, 9\}$ insertarlos, en ese orden, en un *"árbol binario de búsqueda"*. Mostrar las operaciones necesarias para eliminar los elementos 7, 12 y 8, en ese orden.
- **[parcial-3 hash-dict (5 pts)]** Insertar los números $5, 18, 28, 11, 10, 38, 23, 7, 30$ en una tabla de dispersión cerrada con $B = 10$ cubetas, con función de dispersión $h(x) = x\,\%10$ y estrategia de redispersión lineal.

[Ej. 4] [preguntas (10 pts, 2.5 por pregunta)]

a) ¿Cuál es el criterio para elegir una buena función de dispersión? ...

- ☐ Debe tratar de concentrar los elementos en pocas cubetas.
- ☐ Debe tratar de concentrar los elementos en una sóla cubeta.
- ☐ Debe tratar de concentrar los elementos en la primera cubeta.
- ☐ Debe distribuir los elementos en la forma más uniforme posible entre las cubetas.

---

b) Dado el árbol binario `(x e (d f g))`, ¿cuál de las siguientes opciones es verdadera?

- ☐ Es completo y es lleno.
- ☐ Es completo pero no lleno.
- ☐ Es lleno pero no completo.
- ☐ Ni es completo ni es lleno.

---

c) ¿Cuál es el número de niveles en un árbol binario lleno en función del número n de nodos en el árbol?

- ☐ ... $O(n \log n)$
- ☐ ... $O(1)$
- ☐ ... $O(n)$
- ☐ ... $O(\log n)$

---

d) Sea un tabla de dispersión abierta con B cubetas y n elementos. Asumiendo que la función de dispersión es lo suficientemente buena como para distribuir los elementos en forma uniforme entre las cubetas, el costo medio de inserción de un nuevo elemento es

- ☐ $O(n^2/B)$
- ☐ $O((n/B)^2)$
- ☐ $O(n + B)$
- ☐ $O(1 + n/B)$
