

Algoritmos y Estructuras de Datos. Examen Final. [24 de Febrero de 2005]

[Ej. 1] [Clases (20 puntos)]

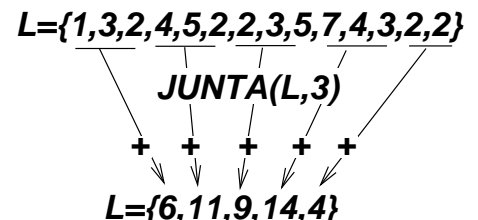
Escribir los métodos `insert(x)`, `erase(iterator p)`, `erase(x)` de correspondientes al TAD conjunto (`class set`) implementado por ABB.

[Ej. 2] [Programación (total = 50 puntos)]

a) [junta (30 puntos)]

Escribir una función `void junta(list<int>&L, int n);` que dada una lista `L`, agrupa de `n` elementos dejando su suma (ver figura).

Restricciones: No usar ninguna estructura auxiliar. Prestar atención a no usar posiciones inválidas después de una supresión. El algoritmo debe tener un tiempo de ejecución $O(n)$, donde n es el número de elementos en la lista original.



b) [maximo-par (10 puntos)]

Escribir una función `int maximo_par(btrees<int>&T);` que retorna el máximo de las etiquetas pares de un árbol binario. En el caso del árbol (10 21 (6 (11 12 6) (23 . 3))) debe retornar 12.

Restricciones: El algoritmo debe tener un tiempo de ejecución $O(n)$, donde n es el número de elementos en el árbol.

c) [elimina-valor (10 puntos)]

Escribir una función `void elimina_valor(queue<int>&C, int);` que elimina todas las ocurrencias del valor `n` en la cola `C`. Por ejemplo, si `C = {1, 3, 5, 4, 2, 3, 7, 3, 5}`, después de `elimina_valor(C, 3)` debe quedar `C = {1, 5, 4, 2, 7, 5}`. *Sugerencia:* Usar una estructura auxiliar lista o cola.

Restricciones: El algoritmo debe tener un tiempo de ejecución $O(n)$, donde n es el número de elementos en la cola original.

[Ej. 3] [operativos (total = 20 puntos)]

a) [rec-arbol (5 ptos)] Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son

- `ORD_PRE = {C, D, E, R, S, Q, A, B}`,
- `ORD_POST = {D, Q, R, A, B, S, E, C}`.

b) [huffman (5 ptos)] Dados los caracteres siguientes con sus correspondientes probabilidades, contruir el código binario y encodar la palabra PIQUETERO

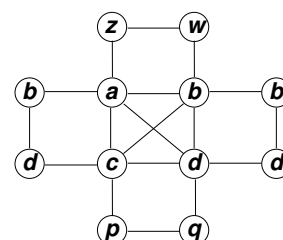
$P(P) = 0,2, P(I) = 0,2, P(Q) = 0,2, P(U) = 0,1, P(E) = 0,1, P(T) = 0,1, P(R) = 0,05, P(O) = 0,05$
 Calcular la longitud promedio del código obtenido.

c) [misc-arbol (5pt)]: Dado el árbol (p (q r s) (t (u v))),

- 1) Cuál es el nodo que está a la vez a la izquierda de `v` y a la derecha de `r`?
- 2) Particione el árbol con respecto al nodo `q`, es decir indique cuales son sus antecesores y descendientes propios, derecha e izquierda

d) [colorear-grafo (5 ptos)]

Colorear el siguiente grafo, utilizando una estrategia heurística para tratar de usar el menor número de colores posibles.



Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]

[Ej. 4] [Preguntas (total = 10 puntos, 2.5puntos por pregunta)] Responder según el sistema “multiple choice”, es decir marcar con una cruz el casillero apropiado. **Atención:** Algunas respuestas son intencionalmente “descabelladas” y tienen puntajes **negativos!!**]

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

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

¿Cuál es el tiempo de ejecución para la función `p--` en listas *doblemente* enlazadas por punteros o cursores?

- ☐ $O(n)$
- ☐ $O(1)$
- ☐ $O(n^2)$
- ☐ $O(\log_2 n)$

Sea L una lista conteniendo los elementos $(4, 7, 6, 5, 3,)$. Después de aplicar las siguientes líneas

```
list<int>::iterator p,q;
```

```
p = L.begin();
```

```
q = ++p;
```

```
p = ++q;
```

```
p = L.erase(p);
```

¿Cuál de las opciones es verdadera?

- ☐ `*p=*q=7.`
- ☐ `*p=5, *q=7.`
- ☐ `*p=5, q` es inválido.
- ☐ `p` y `q` son inválidos.

¿Como afectan los elementos suprimidos (`deleted`) a la eficiencia en una tabla de dispersión cerrada? Es decir, ¿Como es la eficiencia de una tabla A con 50 % indefinidos y 40 % suprimidos con respecto a una tabla B con 50 % indefinidos y sin suprimidos?

- ☐ Equivalen a un elemento ocupado (por lo B es más eficiente que A).
- ☐ Equivalen a un elemento indefinido (`undef`) (por lo tanto B es menos eficiente que A).
- ☐ La eficiencia de A es igual a la de B .
- ☐ No puede haber elementos suprimidos en una tabla de dispersión.