

Algoritmos y Estructuras de Datos. 3er Parcial. Tema: 1A. [22 de noviembre de 2007]

[Ej. 1] [clases (25 puntos)]

- Escribir la implementación en C++ del TAD Conjunto (clase `set`). Implemente: `insert(x)`, `find(x)`, `clear()` y `empty()`. **Observaciones:** Debe declarar los miembros privados de las clases a declarar o implementar. También debe implementar todo método o función auxiliar que necesite. Puede elegir entre implementaciones basadas en *vectores de bits*, *listas ordenadas*, *árboles binarios de búsqueda* y *tablas de dispersión*.
- Implementar una función
`btree<int>::iterator abb_find(btree<int> &T, btree<int>::iterator n, int x)`
 que busca en el ABB T el valor x devolviendo su posición o `T.end()` si no se encuentra.

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

a) [includes-all (20 puntos)]

Dados n conjuntos de vector de conjuntos `vector<set<int>>` A se pide escribir una función `int includes_all(vector< set<int> > &setv);` tal que si alguno de esos conjuntos incluye a todos los otros, es decir existe algún j tal que $A_{[k]} \subset A_{[j]}$, para todo k , entonces debe retornar j . En ese caso contrario, debe retornar -1. Por ejemplo, si $A_{[0]} = \{1, 2, 3\}$, $A_{[1]} = \{1, 2, 3, 4, 5\}$, $A_{[2]} = \{3, 4, 5\}$, entonces debe retornar 1. Si $A_{[0]} = \{1, 2, 3\}$, $A_{[1]} = \{3, 4, 5\}$ entonces debe retornar -1. *Nota:* Si existen varios $A_{[j]}$ que satisfacen la condición entonces debe retornar uno cualquiera de ellos. *Sugerencia:* Notar que, si alguno de los $A_{[j]}$ satisface la condición entonces debe ser el mayor de todos (en cantidad de elementos).

b) [mklayers (30 puntos)]

Escribir una función

`void mklayers(vector<set<int> > &G, int x, vector<set<int> > &layers)` que dado un grafo G y un vértice de partida x determina la estructuras de capas de vecinos layers de x definida de la siguiente manera:

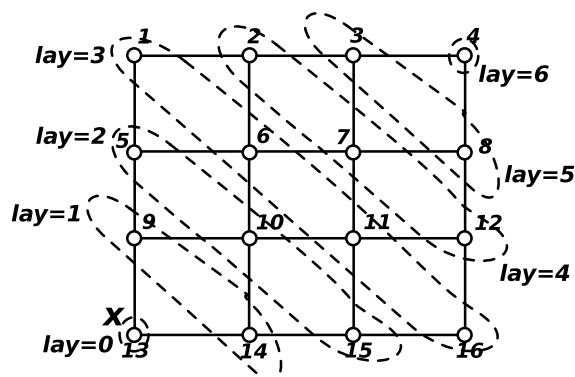
- La capa 0 es el conjunto `layers[0]={x}`.
- La capa 1 es el conjunto de los vecinos de x.
- Para $l > 1$ la capa l es el conjunto de los vecinos de los nodos en la capa l-1 que no están en capas anteriores (0 a l-1). Notar que en realidad sólo hace falta verificar que no estén en las capas l-1 y l-2.

Puede demostrarse que los vértices en la capa l son los que están a distancia l de x. Por ejemplo, dado el grafo de la figura, y partiendo del nodo $x=0$ las capas son
`layers[0]={0}`, `layers[1]={9,14}`, `layers[2]={5,10,15}`, `layers[3]={1,6,11,16}`,
`layers[4]={2,7,12}`, `layers[5]={3,8}`, `layers[6]={4}`.

Apellido y Nombre: _____

Carrera: _____ DNI: _____

[Llenar con letra mayúscula de imprenta GRANDE]



[Ej. 3] [operativos (total = 15 puntos)]

- a) [abb (7.5 pts)] Dados los enteros $\{12, 6, 19, 1, 2, 9, 4, 3, 0, 11\}$ insertarlos, en ese orden, en un “árbol binario de búsqueda”. Mostrar las operaciones necesarias para eliminar los elementos 12, 4 y 11 en ese orden.
- b) [hash-dict (7.5 pts)] Insertar los números 2, 15, 25, 8, 7, 35, 17, 4, 27 en una tabla de dispersión cerrada con $B = 8$ cubetas, con función de dispersión $h(x) = x \bmod 8$ y estrategia de redispersión lineal.

[Ej. 4] [Preguntas (total = 10 puntos, 3.33 puntos 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!!**

- a) ¿Cuál es el tiempo de ejecución de $\text{find}(x)$ en el TAD diccionario por tablas de dispersión abiertas, en el caso promedio?

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

- b) ¿Cual es el tiempo de ejecución para $\text{insert}(x)$ en el TAD conjunto por árbol binario de búsqueda, en el peor caso?

- ☐ ... $O(n)$
☐ ... $O(n^2)$
☐ ... $O(1)$
☐ ... $O(\log n)$

- c) ¿Cual es el tiempo de ejecución para $\text{insert}(x)$ en el TAD conjunto por tabla de dispersión cerrada, en el caso promedio?

- ☐ ... $O(n^2)$
☐ ... $O(1)$
☐ ... $O(\log n)$
☐ ... $O(n)$