

Nota important: Es valorarà molt l'elegància i simplicitat de les solucions aportades.

Problema 1 Miscel·lània de conjunts (4 punts)

La classe *ConjuntStrings*<*N*> és la classe dels conjunts de cadenes de vocals ('a', 'e', 'i', 'o', 'u') que tenen un màxim de *N* caràcters.

Volem representar aquesta classe mitjançant un *vector de bits* de la manera següent:

```
template<unsigned int N>
class ConjuntStrings{

    bool v[R];

public:
    ConjuntStrings();
    void inserir (char* st);
    void eliminar (char* st);
    bool pertany (char* st);
    friend ConjuntStrings operator+(ConjuntStrings& c1,
                                    ConjuntStrings& c2);
    friend ConjuntStrings operator*(ConjuntStrings& c1,
                                    ConjuntStrings& c2);
};
```

la idea d'aquesta representació és que un conjunt contindrà la cadena *s* si i només si $v[f[s]] = cert$; on *f* és una funció que transforma les cadenes de caràcters en naturals a partir de 0 i assegurant que dues cadenes diferents generaran sempre valors diferents de *f* (*f* aplicació injectiva)..

Es demana:

1. (0,5 punts) Quant ha de valdre *R* (evidentment, cal posar-ho en funció de *N*).
2. (0,5 punts) Escriure una possible funció *f* en pseudocodi. Quin és el cost de *f*?
3. (1 punt) Volem comparar el cost temporal de diferents representacions de conjunts de strings. Omplir la següent taula comparativa de costos. Si heu d'afegir algun comentari aclaridor feu-ho a

la taula de sota.

Num	Representació	inserir	eliminar	pertany	unio(+)
1	Vector de bits				
2	Llista implementada en mem. dinàmica amb apuntadors				
3	Taula implementada com a taula de dispersió				
4	Arbre binari de cerca				
5	Vector ordenat				

Num	Comentari
1	
2	
3	
4	
5	

4. (2 punts) En C++ el tipus `bool` es representa amb un byte (tot i que amb un bit n'hi hauria prou). Així doncs, la declaració:

`bool v[R];` (amb R que s'ha calculat anteriorment) no aprofita bé l'espai.

Volem representar la classe `ConjuntString` com un vector de bits però sense malbaratar tots els bits que no s'utilitzen amb la representació habitual. Quina hauria de ser R ara? Quina seria la funció f ? Quins serien els codis de *pertany* i *unio*?

Problema 2 Iteradors sobre arbres binaris en inordre (3,5 punts)

1. Representa la classe $ABin<T>$ amb apuntador al pare. Només cal que descriguis la part privada de la classe. Dibuixa la representació en memòria d'un arbre binari buit i un arbre binari amb 4 nivells usant la representació que has proposat prèviament.
2. Implementa les operacions *situarInici*, *operator++* (prefix) i *fi* per la classe *IteABinIn* (això és, iteradors sobre arbres binaris en inordre). Suposa que la classe $ABin<T>$ està implementada tal com s'ha indicat a l'apartat anterior.

Problema 3 Els vectors excepcionals (3,5 punts)

Especifica i implementa la classe *Vector* genèrica dotada d'una excepció *Rang* que s'activi en el moment que s'intenti accedir a un índex no definit pel vector. Aquesta excepció haurà d'indicar a quin índex fora de rang s'ha intentat accedir. La classe *Vector* haurà d'implementar les operacions següents:

- $Vector<T>(int\ cap)$; Construïm un vector amb capacitat *cap*
- $void\ redefinirCapacitat(int\ c)$; *c* és la nova capacitat del vector. No s'han de produir deixalles.
- Sobrecàrrega de l'operador $[]$ de manera que es puguin fer crides de la forma: $v[i] = x$; on *v* es un objecte de la classe $Vector<T>$.
- Sobrecàrrega de l'operador $<<$
- Quines hipòtesis s'han de fer sobre el tipus que instanciarà *T* a $Vector<T>$? Afegeix aquestes hipòtesis a l'especificació de la classe $Vector<T>$.