

Titulació: Grau en Enginyeria Informàtica
Assignatura: Programació 2 (PRO2)
Duració: 2h 30m

Curs: Q1 2019–2020 (2n Parcial)
Data: 13 de Gener de 2020

1. (5 punts) Considerem la següent definició:

```
class MySet {
public:
    ...
    // Pre: cert
    // Post: retorna un MySet amb la subllista
    //        formada pel 1r, 2n, 4r, 8è, 16è, ...
    //        elements del MySet implícit
    MySet galopar() const;

private:
    // els elements del MySet està emmagatzemats
    // en una llista encadenada simple
    struct node {
        int info;
        node* seg;
    };
    node *primer, *ult; // apuntadors al primer i a l'últim
                        // element de la llista
}
```

La classe `MySet` està implementada mitjançant una llista **simplement** encadenada amb apuntadors al primer i a l'últim element de la llista. L'últim element d'un `MySet` no té successor; si `ult != nullptr` llavors hi ha un últim element i `ult -> seg == nullptr`. En un `MySet` buit `primer == ult == nullptr`. Si el `MySet` conté només un element llavors `primer == ult` apunten a l'únic node de la llista.

- (a) (2 punts) Implementa un mètode `static` privat que donat un apuntador `p` a un node d'una cadena de nodes i un valor enter no negatiu `k` retorna un apuntador al node que està `k` posicions més enllà de `p` a la cadena o bé `nullptr` si no hi ha `k` o més nodes a continuació de `p` a la cadena.

```
// Pre: p ≠ nullptr, k ≥ 0
// Post: retorna un apuntador al node que està k posicions més
//        enllà de p a la cadena o bé nullptr si no hi ha k o més nodes
//        a continuació de p
static node* salta(node* p, int k);
```

- (b) (3 punts) Implementa el mètode `galopar`. Pots fer servir el mètode privat de l'apartat anterior, i d'altres si vols, sempre i quan els especifiquis i implementis completament.

2. (5 punts) Considerem la següent definició d'una classe `ArbreBinari` en C++

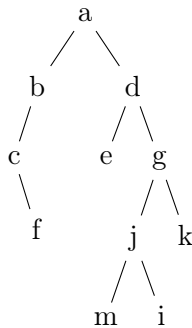
```
template <typename T>
class ArbreBinari{
private:
    struct node {
        T info;
        node* esq;
        node* dre;
    };
    node* arrel; // apuntador a l'arrel de l'arbre
    ...
public:
    ...
    double area() const; // calcula l'àrea del paràmetre implícit
    ...
};
```

S'anomena *àrea* d'un arbre a a la suma dels ratios entre el nombre de nodes a cada nivell d' a i el nombre màxim de nodes que pot haver-hi a cada nivell. Si $N_k(a)$ denota el nombre de nodes de l'arbre a en el nivell $k \geq 0$ llavors

$$S(a) = \sum_{k \geq 0} \frac{N_k(a)}{2^k} \quad (*)$$

Recorda que si un arbre no és buit la seva arrel està al nivell 0, els fills de l'arrel estan a nivell 1, etc. Com que $N_j(a) = 0$ a partir d'un cert nivell j , el sumatori donat just abans consta d'un nombre finit de termes i està ben definit.

Per exemple, l'arbre



té àrea

$$1 + 2/2 + 3/4 + 3/8 + 2/16 = 3.25.$$

(a) (1 punt) Demostra que la següent definició **recursiva**

$$S(a) = \begin{cases} 0 & \text{si } a = \text{ArbreBinari}(), \\ 1 + \frac{1}{2} (S(\ell) + S(r)) & \text{si } a = \text{ArbreBinari}(x, \ell, r), \end{cases}$$

és equivalent a la definició donada per la equació (*).

- (b) (3 punts) Especifica i implementa en C++ un o més mètodes `static` privats de la classe `ArbreBinari` que ens permetin calcular eficientment l'àrea d'un arbre binari. Pots utilitzar la definició de l'apartat anterior.
- (c) (1 punt) Implementa el mètode públic `area`, usant el o els mètodes privats definits a l'apartat anterior.