

Cognoms

Nom

DNI

Problema 1 (5 punts)

Donat un vector v d'enters **estrictament positius** es demana comprovar si v és pot dividir en tres segments no buits (tenen al menys un element) tal que sumin el mateix. En cas de que existeixin s'ha de tornar l'índex del primer y el darrer element del segment central, en cas contrari s'ha de tornar 0 y $v.size()-1$ (el primer i el darrer índex del vector). Per simplificar el codi, hi ha un paràmetre que conté la suma dels elements del vector.

1.1 (2.5 punts)

Implementeu **de forma eficient** la següent acció:

```
void tres_segments (const vector<int> &v, int suma, int &i, int &j) {  
    /* Pre: v.size() ≥ 3, tots els elements de v són estrictament positius,  
    suma = suma(v[0..v.size()-1]) */  
    /* Post: i i j tal que la suma de v[0..i-1] i la suma de v[j+1..v.size()-1]  
    i la suma de v[i..j] són iguals, 0 i v.size()-1 si v  
    no es pot partir en tres segments no buits que sumin igual */
```

Pistes: És molt important tenir en compte que tots els elements són majors que zero. Si s'aconsegueix que els segments dels extrems sumin igual, es pot mirar si a més el segment central també suma igual.

1.2 (2.5 punts)

Per a tots els bucles que implementeu: a) proposeu un invariant, b) proposeu una funció de fita, c) justifiqueu la inicialització, d) justifiqueu la condició de sortida, e) justifiqueu el cos del bucle i f) justifiqueu l'acabament.

Cognoms

Nom

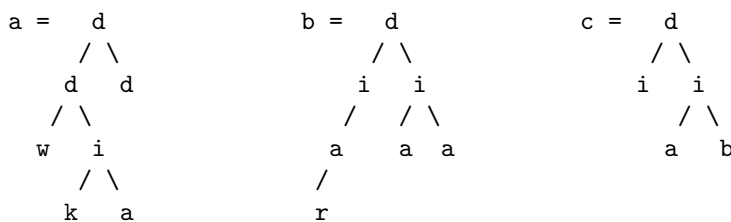
DNI

Problema 2 (5 punts)

Direm que un `BinTree<char>` *a* representa una paraula `pal` formada pels caràcters $c_1c_2 \dots c_n$ amb $n > 0$ si hi ha un camí a l'arbre començant des de l'arrel i de longitud n tal que els valors dels nodes del camí formen `pal`. Implementeu eficientment la funció `quants_cops`, especificada a continuació:

```
int quants_cops(const BinTree<char> & a, const vector<char> & pal);
/* Pre: pal té al menys un caràcter */
/* Post: el resultat és quants cops a representa pal */
```

Per exemple, per la paraula `dia` si l'arbre és *a*, el resultat és 0, si l'arbre és *b* el resultat és 3 i si l'arbre és *c* el resultat és 1.



2.1 (2.5 punts)

Dissenyeu una funció recursiva d'immersió donant: a) la seva capçalera i especificació, b) la seva implementació i c) la implementació de la funció original.

2.2 (2.5 punts)

Justifiqueu la correctesa del vostre codi donant: d) una funció de mida, e) les hipòtesis d'inducció, f) justificació del seu acabament, g) justificació del(s) cas(os) base i h) justificació dels cas(os) recursiu(s).