

ALEXANDRE SÁNCHEZ CASALS

```
#include <iostream>
#include <list>
#include <vector>
using namespace std;

typedef list<int>::iterator it;

void trams(const list<int>& l, vector<list<int> >& v){
/* Pre: v.size() == 0, */
/* Post: la mida de v és el nombre de trams de l,
       cada posició v[i] conté una llista amb l'i-èssim tram de l */

    it1 = l.begin();
    it2 = l.begin();
    ++it2;
    int i = 0;
    bool btram = false;
    while (it2 != l.end()){
        list<int> tram;
        it aux = tram.begin();
        if((*it1) <= (*it2)){
            tram.insert(aux,it1);
            ++it1;
            ++it2;
            btram = true;
        }
        else {
            if (btram){
                tram.insert(aux,it1);
                btram = false;
                ++it1;
                ++it2;
            }
            else {
                ++it1;
                ++it2;
            }
        }
        if (it2 == l.end()){
            --it1;
            --it2;
            if((*it2) >= (*it1)){
                tram.insert(aux,it2);
            }
            else {
                list<int> ultim(1);
                it it3 = ultim.begin();
                ultim.insert(it3,(*it2));
                ++it1;
                ++it2;
            }
            btram = true;
        }
        v[i] = tram;
        ++i;
        if(!((*it2) >= (*it1))){
            v[i] = ultim;
        }
    }
}
```

/*

INVARIANT:

- it2 es troba entre l.begin() i l.end(), it1 apunta a la posició anterior a it2 i acaba en la posició anterior a l.end().
- v conté els trams des de [l.begin()...it2].
- btram indica si els elements als que apunten it1 i it2 formen part d'un tram.
- i indica la posició entre v[0...v.size()-1].

CONDICIÓ DE SORTIDA:

- Ens mantindrem dins el bucle (while) mentre l'iterador it2 sigui diferent de l.end()

COS DEL BUCLE:

- Avançem els iteradors, i discernim si l'element al que es troba it1 forma part d'un tram, per afegir-lo a la llista tram.
Si it2 apunta a l.end(), pot ser que l'últim element (al que apunta it1) formi part d'un tram o ell conformi un tram per si sol, per això movem els iteradors a la posició anterior i, o bé l'insertem al tram actual (list tram) o creem una llista amb un sol element que es correspongui a l'últim.

ACABAMENT:

- it1 apunta a l'últim element de la llista
- it2 apunta a l.end();
- btram és true, ja que o bé l'últim element forma part d'un tram de més d'un element, o ell és un tram d'un element.
- i indica l'última posició del vector v[v.size()-1].

*/