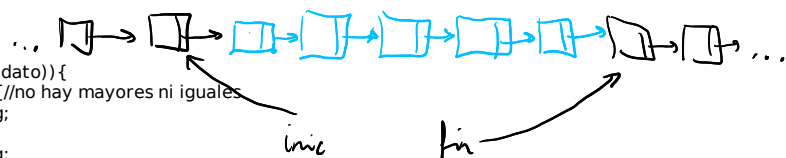


```
template <typename T>
void pivotar(Sequencia<T> s, Nodo_sec<T>* inic, Nodo_sec<T>* fin, T pivote, Nodo_sec<T>* &ult_menor, Nodo_sec<T>* &ult_igual){
    ult_igual = inic;
    ult_menor = inic;
    Nodo_sec<T>* ult_mayor = inic;
    Nodo_sec<T>* aux;
    while(ult_mayor->sig != fin){
        if(!(pivote<=ult_mayor->sig->dato)){
            if(ult_mayor == ult_menor){//no hay mayores ni iguales
                ult_menor = ult_mayor->sig;
                ult_igual = ult_mayor->sig;
                ult_mayor = ult_mayor->sig;
            }
            else{
                aux = ult_mayor->sig;
                ult_mayor->sig = aux->sig;
                aux->sig = ult_mayor->sig;
                if(ult_igual == ult_menor){//no hay iguales
                    ult_menor->sig = aux;
                    ult_menor = aux;
                    ult_igual = aux;
                }
                else{
                    ult_mayor->sig = aux;
                    ult_mayor = aux;
                }
            }
        }
        else if(!(ult_mayor->sig->dato<=pivote)){
            ult_mayor = ult_mayor->sig;
        }
        else{
            if(ult_mayor == ult_igual){//no hay mayores
                ult_mayor = ult_mayor->sig;
                ult_igual = ult_mayor->sig;
            }
            else{
                aux = ult_mayor->sig;
                ult_mayor->sig = aux->sig;
                aux->sig = ult_mayor->sig;
                ult_igual->sig = aux;
                ult_igual = aux;
            }
        }
    }
}
```

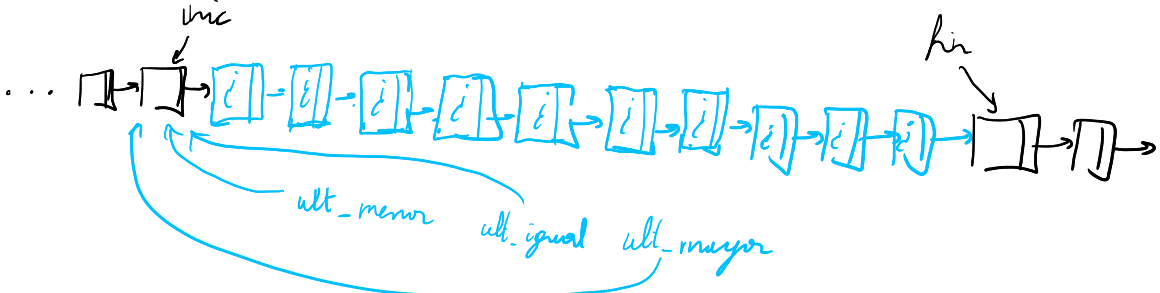


Reordena el "buzo azul" poniendo primer los menores que pivote, a continuación los iguales y después los mayores

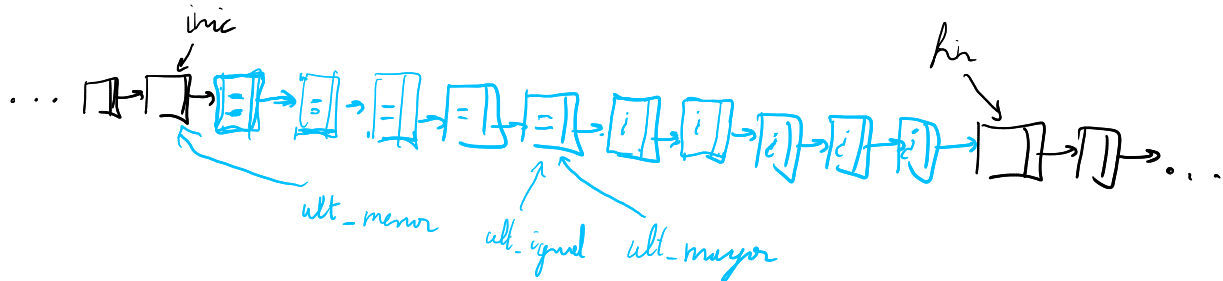
Situaciones durante el bucle:



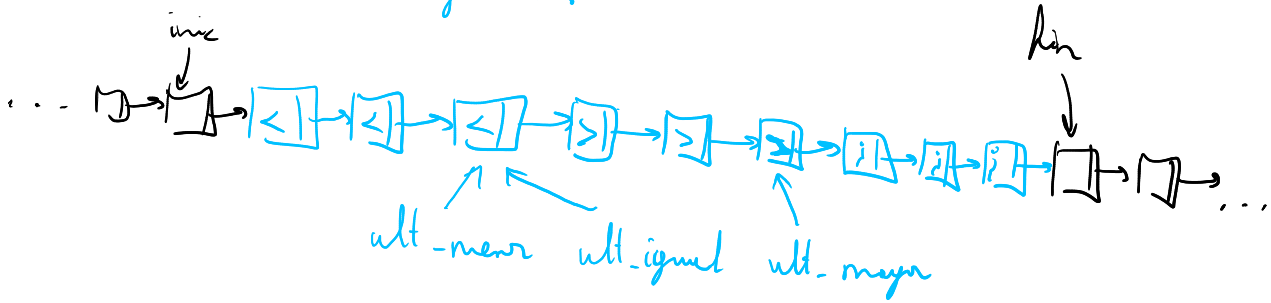
al inicio:



sin menores, ni mayores:



sin iguales:

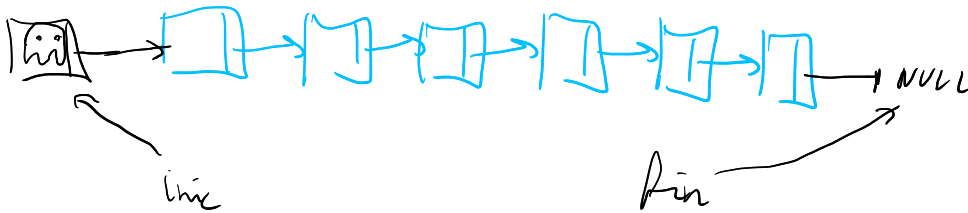


```

template <typename T> // T debe tener definido el operador <=
    //ordena la secuencia s
void quick_sort_in_situ(Secuencia<T> s, Nodo_sec<T>* inic, Nodo_sec<T> *fin){
    if(inic->sig!=fin){
        Nodo_sec<T> *ult_menor, *ult_igual;
        T p = inic->sig->dato;
        pivotar(s,inic,fin,p,ult_menor,ult_igual);
        quick_sort_in_situ(s,inic,ult_menor->sig);
        quick_sort_in_situ(s,ult_igual,fin);
    }
}

```

de llamada para ordenar toda la secuencia:



`quick_sort_in_situ (s, s.primer, (Nodo_sec<T> *) NULL)`