

Capitolo 8

Esercizio (5)

Esercizio (5) del Capitolo 8

A ha top elementi, $\text{top} \geq 0$, ed è non decrescente

$A = [p_1 \dots p_k, u_1 \dots u_q, g_1 \dots g_w] \rightarrow [p_1 \dots p_k, g_1 \dots g_w]$

$\text{top} = k + q + w$

$u_1 = y, \dots, u_q = y$

per ogni i in $[1..k]$ $p_i < y$

per ogni i in $[1..w]$ $g_i > y$

visto che top potrebbe anche essere 0, k , q e w possono essere tutte 0 ed in ogni caso ciascuna di esse può essere 0.

Importante: individuare operazioni più semplici

- a) se c'è y , in che indice inizia e quanti y ci sono
 - i) determinare se c'è un y e la posizione del primo y
 - ii) determinare quanti y ci sono-
- b) eliminare gli y (se ci sono)

compito (a)

PRE=(A=[p1..pk,u1..uq,g1..gw], k+q+w=top>=0)

bool trova(int*A,int top,int y, int&start, int&quanti)

POST=(trovato=> start=k, quanti=q) &&
(!trovato=> A[0..top-1] non contiene y)

```

bool trova( int*A, int top, int y, int&start,int&quanti)
{
    bool trovato=false;
    for(int i=0; i<top && A[i]<=y && !trovato; i++) //R1
        if(A[i]==y)
            {trovato=true; start=i;}
    //(trovato=> start=k ) && (!trovato => A[0..top-1]!=y)

    if(trovato)
    {
        quanti=1;
        for(int i=start+1; i<top && A[i] ==y; i++) //R2
            quanti++;
        } // (trovato=>quanti=q)
    return trovato;
}

```

Compito (b): eliminare le y

PRE(A[0..top-1] definito, $0 \leq \text{start} \leq \text{start} + \text{quanti} - 1 < \text{top}$) && (vA e vtop sono valore iniz. di A e top)

void shift(int*A, int&top, int start, int quanti)

POST=(A[0..start-1]=vA[0..start-1]) &&
(A[start..vtop-quanti-1]=vA[start+quanti..vtop-1]) &&
(top=vtop-quanti)

```
void shift(int*A, int&top, int start, int quanti)
{
    for(int i=start; i<top-quanti; i++)
        A[i]=A[i+quanti];

    top=top-quanti;
}
```

```
void F(int* A, int&top,int y)
{
    int start, quanti;
    if(cerca(A,top,start,quanti))
        shift(A,top,start,quanti);
}
```