Il problema della bandiera tricolore.

Dato un vettore di elementi colorati in uno dei tre colori della bandiera italiana (dove elementi di uno stesso colore possono essere fra loro diversi per altre caratteristiche), si ordini parzialmente il vettore in modo che tutti gli eventuali elementi verdi precedano tutti gli eventuali elementi bianchi i quali a loro volta precedano tutti gli eventuali elementi rossi, in tempo lineare e spazio costante (cioè senza usare un vettore ausiliario). Ossia:

SITUAZIONE FINALE:

$x \in v[1..i-1] \rightarrow$ x è verde

$x \in v[i..j-1] \rightarrow$ x è bianco

$x \in v[j..n] \rightarrow$ x è rosso

(dove $1 \leq i \leq j \leq n+1$)

Soluzione

```
const n = ...
type italcolore = (verde, bianco, rosso);
     elemcolorato = record
                      nome: ...;
                      colore: italcolore;
                    end;

     bandiera = array[1..n] of elemcolorato;

procedure dividicolori(var v: bandiera);
var i,j,k: integer;
begin
  i:= 1; j:= 1; k:= n;
  while j<=k do
    if a[j].colore = verde then begin
      scambia(v[i],v[j]);
      inc(i); inc(j)
    end
    else if v[j].colore = bianco then inc(j)
    else {v[j].colore = rosso} begin
      scambia(v[j],v[k]);
      dec(k)
    end
end;
```

Dimostrazione di correttezza per la procedura della bandiera tricolore.

Indichiamo con INV la seguente proposizione:

INV:

$$
\begin{aligned}
&\mathbf{1 \le i \le j \le k{+}1 \quad \wedge \quad k \le n} \\
&\mathbf{\wedge\ x \in v[1..i\text{-}1] \quad \rightarrow \quad x\ \text{è verde}} \\
&\mathbf{\wedge\ x \in v[i..j\text{-}1] \quad \rightarrow \quad x\ \text{è bianco}} \\
&\mathbf{\wedge\ x \in a[k{+}1..n] \quad \rightarrow \quad x\ \text{è rosso}}
\end{aligned}
$$

dove il simbolo $\wedge$ si legge "e" (congiunzione logica), e $A \rightarrow B$ si legge "se A, allora B".

Teorema 1.
INV è vera immediatamente dopo qualunque numero H (aanche se 0) di esecuzioni del corpo del ciclo (cioè INV è l'*invariante* del ciclo).

Dimostrazione per induzione.

Base.
INV è vera dopo 0 esecuzioni del corpo del ciclo, cioè:
INV è vera nell'istante precedente l'esecuzione dell'istruzione *while* (cioè prima di entrare per la prima volta nel ciclo).

Dimostrazione della base.
Sono state eseguite le istruzioni di inizializzazione, e si ha quindi $i=1$, $j=1$, $k=n$.
Si vede facilmente che sostituendo tali valori a i,j,k la prima riga di INV è soddisfatta, e le tre righe successive sono banalmente soddisfatte perché i tre segmenti di vettore diventano v[1..0], v[1..0], v[n=1..n] e sono quindi vuoti.

Passo induttivo.
Ipotesi (induttiva):
Dopo H esecuzioni del corpo del ciclo INV è vera <u>e il test del *while* è vero</u>, cioè:
dopo H esecuzioni del corpo del ciclo si ha INV $\wedge$ TEST, cioè:

Dopo H esecuzioni del corpo del ciclo si ha INV $\wedge\, j \le k$.

Tesi (induttiva):
Dopo H+1 esecuzioni del corpo del ciclo INV è vera, cioè:

Dopo H esecuzioni del corpo del ciclo si ha INV.

Dimostrazione del passo.
Si osservi che, per l'ipotesi induttiva, dopo la H-esima esecuzione INV è vero e quindi si ha $i \le j$; consideriamo la H+1-esecuzione nei tre casi: se v[j] è verde vengono incrementati i e j e quindi sarà ancora $1 \le i \le j$; se v[j] è bianco viene incrementato solo j e quindi sarà ancora $1 \le i \le j$; se v[j] è rosso i e j non vengono modificati; si è così dimostrato che alla fine dell'esecuzione si ha ancora, in tutti i casi, $1 \le i \le j$.

Si osservi ora che, per l'ipotesi induttiva, dopo l'H-esima esecuzione non solo vale INV, ed è quindi $j \le k+1$, ma vale anche TEST, cioè vale la condizione più forte $j \le k$; allora consideriamo di nuovo l'esecuzione nei tre casi: se v[j] è verde oppure bianco j viene incrementato e k no, e quindi può darsi che dopo non sia più $j \leq k$, ma sarà certamente $j \leq k+1$; se v[j] è rosso j non viene modificato ma k viene decrementato, quindi può darsi che non sia più $j \leq k$, ma sarà certamente $j \leq k+1$; abbiamo così dimostrato che alla fine della H+1-esima iterazione si ha di nuovo $j \leq k+1$ in tutti i casi.

Si osservi infine che per l'ipotesi induttiva è $k \leq n$, ma nel corpo del ciclo k o non viene modificata o viene decrementata, quindi dopo l'esecuzione sarà ancora $k \leq n$.

Abbiamo così dimostrato dopo la H+1-esima iterazione la prima riga di INV è ancora vera.

Per quanto riguarda le tre righe successive di INV, basta esaminare di nuovo i tre casi considerando, oltre alle variazioni degl'indici, anche le operazioni di scambio che si effettuano in ciascuno dei casi, e si vede facilmente che le tre proposizioni rimangono vere dopo l'esecuzione del corpo del ciclo (le istruzioni di scambio e variazione di indici le abbiamo scritte apposta, si faccia riferimento alla spiegazione intuitiva orale sul disegno).
Abbiamo così dimostrato che l'intera proposizione INV è ancora vera dopo la H+1-esima iterazione.

Fine della dimostrazione del passo.
Fine della dimostrazione del teorema.

Teorema 2.
All'uscita dall'istruzione *while* (cioè all'uscita definitiva dal ciclo) vale la proposizione:

SITUAZIONE FINALE:

$$\begin{array}{lll} & 1 \leq i \leq j \leq n+1 & \\ \wedge\ x \in v[1..i-1] & \rightarrow & x \text{ è verde} \\ \wedge\ x \in v[i..j-1] & \rightarrow & x \text{ è bianco} \\ \wedge\ x \in v[j..n] & \rightarrow & x \text{ è rosso} \end{array}$$

cioè il vettore è ripartito in tre segmenti contigui, alcuni dei quali eventualmente vuoti, nell'ordine verde, bianco, rosso.
Si noti che la condizione sugl'indici (cioè la prima riga della proposizione) asserisce che essi individuano correttamente tre segmenti <u>interamente contenuti</u> nel vettore.

Dimostrazione.
Per il teorema 1, all'uscita dell'istruzione while vale INV; ma, poichè appunto se si è usciti dal while vuol dire che il test del while ha dato risultato false, deve valere anche la negazione del test, cioè $\neg(j \leq k)$, cioè $j>k$:

$$\mathbf{INV} \wedge \neg\mathbf{TEST}$$

cioè

$$\mathbf{INV} \wedge j>k$$

Ma per INV si ha $j \leq k+1$, quindi dalla congiunzione con $j > k$ si deduce che è $j = k+1$ (ricordare che j e k sono interi!), ossia $k = j-1$. Si ha quindi INV $\wedge$ $k = j-1$; sostituendo k con j-1 in INV si ottiene:

$$
\begin{aligned}
&\mathbf{1 \le i \le j \le j\text{-}1\text{+}1 \quad \wedge \quad j\text{-}1 \le n} \\
&\mathbf{\wedge\ x \in v[1..i\text{-}1] \quad \rightarrow \quad x\ \text{è verde}} \\
&\mathbf{\wedge\ x \in v[i..j\text{-}1] \quad \rightarrow \quad x\ \text{è bianco}} \\
&\mathbf{\wedge\ x \in v[j\text{-}1\text{+}1..n] \quad \rightarrow \quad x\ \text{è rosso}}
\end{aligned}
$$

da cui, semplificando, si ha la SITUAZIONE FINALE:

$$
\begin{aligned}
&\mathbf{1 \le i \le j \le n\text{+}1} \\
&\mathbf{\wedge\ x \in v[1..i\text{-}1] \quad \rightarrow \quad x\ \text{è verde}} \\
&\mathbf{\wedge\ x \in v[i..j\text{-}1] \quad \rightarrow \quad x\ \text{è bianco}} \\
&\mathbf{\wedge\ x \in v[j..n] \quad \rightarrow \quad x\ \text{è rosso}}
\end{aligned}
$$

Fine della dimostrazione del teorema 2.

Abbiamo così dimostrato che, <u>se</u> e quando l'esecuzione esce dal *while*, il problema è risolto correttamente.

Dobbiamo però ancora appunto dimostrare che l'esecuzione <u>esce</u> dal *while*, cioè che:

Teorema 3.
Dopo un numero finito di iterazioni la condizione TEST diventa falsa.

Dimostrazione.
Ad ogni iterazione viene o incrementato j (se v[j] è verde oppure bianco), oppure decrementato k (se v[j] è rosso); poichè inizialmente si ha j=1 e k=n, dopo esattamente n passi si avrà j>k.

Tale dimostrazione dimostra quindi, in realtà, un teorema più forte:

Teorema 4.
L'algoritmo termina correttamente <u>e la sua complessità temporale è Θ(n)</u> in ogni caso (non ci sono casi migliore. medio, e peggiore).