Small Examples

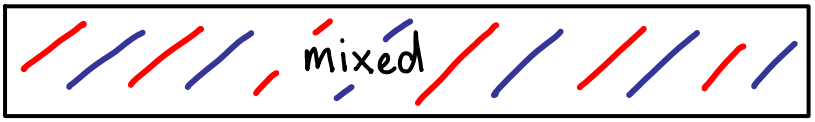
Note Title 01/09/2008

- small illustrations of loop design
  — invariants and making progress
  - simple sorting problem (Dutch National Flag)
  - rotation problem
  - searching problem (saddleback search)

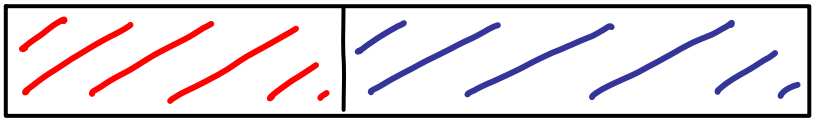
A Sorting Problem

The elements of an array are either red or blue. Construct a program to sort the array so that the red elements form an initial segment of the array.

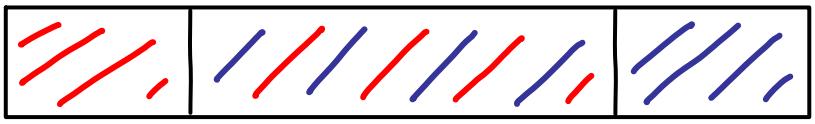
precondition



postcondition



Invariant:



Making progress: reduce size of "mixed" region

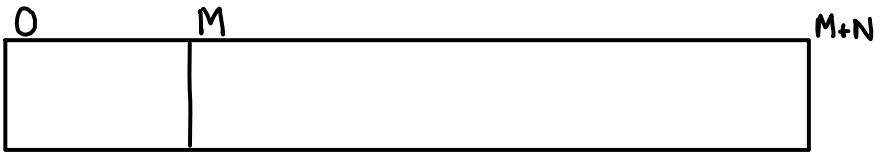
Rotation Problem

An array has length M+N. Construct a program to rotate the array elements so that the initial segment of length M is swapped with the final segment of length N.

precondition

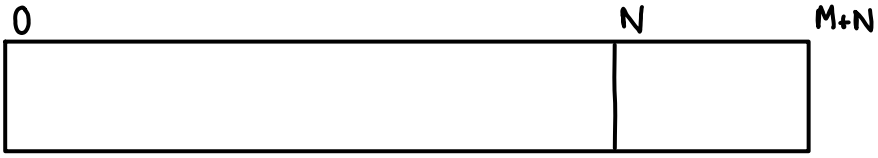
postcondition

precondition

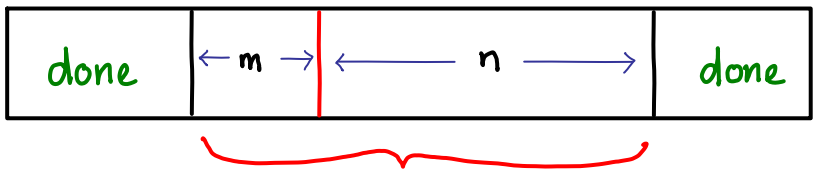


invariant

postcondition



Invariant:



to be rotated
about red line

$\{ a = B + C \wedge \#B = M \wedge \#C = N \}$

initialisation

{ Invariant: $a = u + v + w + x \wedge C = u + w \wedge B = v + x$

Bound fn. : $\#(v + w)$ }

loop

$\{ a = C + B \}$

Assume blockswap(i,j) swaps segment of length i ending at index j-1 with the equal-length segment beginning at index j. That is:

$\{ \quad i \leq j \;\wedge\; j+i \leq M+N \;\wedge$

$a = a_0[0..j-i) +\!\!+ a_0[j-i..j) +\!\!+ a_0[j..j+i) +\!\!+ a_0[j+i..M+N) \}$

blockswap(i,j)

$\{ a = a_0[0..j-i) +\!\!+ a_0[j..j+i) +\!\!+ a_0[j-i..j) +\!\!+ a_0[j+i..M+N) \}$

$\{ 0<M \;\wedge\; 0<N \;\wedge\; a = a_0[0..M) +\!\!+ a_0[M..M+N) \}$

$m,n,k := M,N,M$;

{Invariant: $0<m \;\wedge\; 0<n \;\wedge\; k = N+m-n$

$a = a_0[M..M+N-n) +\!\!+ a_0[0..m) +\!\!+ a_0[M+N-n..M+N) +\!\!+ a_0[m..M)$

Bound fn: $m+n$ }

do $m<n \longrightarrow$ blockswap(m,k) ; $n,k := n-m, k+m$

□ $n<m \longrightarrow$ blockswap(n,k) ; $m,k := m-n, k-n$

od

; $\{ m = M \nabla N = n \}$

blockswap(m,k)

$\{ a = a_0[M..M+N) +\!\!+ a_0[0..M) \}$

Saddleback Search

A two-dimensional array is strictly increasing along rows and along columns. Construct a program to count how many times a given value x occurs in the array.

.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
3	9	13	15	17	23	36	41	42
1	6	8	11	12	22	35	40	41
0	5	7	10	11	21	30	31	32