http://public.gettysburg.edu/~cpresser/cs201/10-19.pdf

Algorithm: getMinimum

precondition: *data* is an array of integers

```
min := data[0]
i := 1

while(i < data.length)  //Guard G
   if(data[i] < min)     //Condition B
      min := data[i]
   end if
   i := i + 1
end while
```

postcondition: *min* ≤ *data*[i] for 0 ≤ i < *data.length*

invariant: prior to the n^{th} iteration.
I(n): i = n and
min ≤ *data*[j] for 0 ≤ j < *n*

I. Basis: invariant is true prior to the first iteration.

$I(1)$:

$i = 1$? ✓ 2nd assignment

$min \leq data[j] \quad 0 \leq j < 1$

$min \leq data[0]$ ✓

II. Induction

Suppose: $G \wedge I(k)$

Show: $I(k+1)$

Suppose: $i < data.length$ and $i = k$ and $min \leq data[j]$ $0 \leq j < k$

Show: $i = k+1$ and $min \leq data[r]$ $0 \leq r < k+1$

show: ~~$i = k+1$~~

from $i := i+1$, $i_{new} = i_{old} + 1$

and $i_{old} = k$

$i_{new} = k+1$

---

show $min \leq data[r]$ $0 \leq r < k+1$

case 1: B is false

so $min \leq data[k]$

since $min \leq data[j]$ $0 \leq j \leq k$ by I.H.

and $min \leq data[k]$

we know $min \leq data[r]$ $0 \leq r < k+1$

case 2: B is true

$min > data[k]$

$min_{new} := data[k]$!

$min_{old} \leq data[j]$ $0 \leq j < k$

$min_{new} < min_{old}$

since min_{old} was smaller than all items up to k, then min_{new} must be as well.

so $min \leq data[j]$ $0 \leq j < k+1$ |

III Eventual falsity of the Guard

false when i = data.length

from I (data.length)

IV Correctness of post condition

Since the guard first becomes false prior to iteration data.length

I (data.length) is true

The post condition comes directly from I (data.length)

Problem: move k disks from position A to position B

Solution

- move k-1 (top) disks from A to C
- move bottom disk to B
- move k-1 disks from C to B

min # of moves to move k disks from A to B = min # to move k-1 from A to C + 1 + min # to move k-1 from C to B