Practice Problems

PRACTICE PROBLEMS

Q4 $Fib(0) = 0$, $Fib(1) = 1$

$Fib(n) = Fib(n-1) + Fib(n-2)$. for $n > 1$

prove by induction: $1.5^n < Fib(n) < 1.7^n$ for $n >= 11$

$[Fib(11) = 89,\ Fib(12) = 144]$

For $n = 11$

$$1.5^{11} = 86.49 < Fib(11) = 89 < 1.7^{11} = 342.72$$

by IH: it is true for $11 \dots n-1 \Rightarrow$ it is true for n.

True for $11 \dots n-1$

$$\Rightarrow \quad 1.5^{n-1} < Fib(n-1) < 1.7^{n-1}$$

$$\& \quad 1.5^{n-2} < Fib(n-2) < 1.7^{n-2}$$

$+$.

$$1.5^{n-2}\underbrace{(1.5+1)}_{2.5} < Fib(n-1) + Fib(n-2) < 1.7^{n-2}\underbrace{(1.7+1)}_{2.7}$$

$$1.5 \times 1.5 = 2.25 < 2.5.$$

$$\Rightarrow \quad 1.5^{n-2}(2.5) > 1.5^n$$

~~Similarly~~ And

$$1.7 \times 1.7 = 2.89 > 2.7.$$

$$\Rightarrow \quad 1.7^{n-2}(2.7) < 1.7^n$$

$$\Rightarrow \quad 1.5^n < \underbrace{Fib(n-1) + Fib(n-2)}_{} < 1.7^n$$

$$\Rightarrow \quad 1.5^n < Fib(n) < 1.7^n.$$

Proved

Q5 Practice Problems

Prove that $\sum_{i=0}^{n} Fib(i) = Fib(n+2) - 1$

for $n = 0$

$$Fib(0) = 0$$

$$\sum_{i=0}^{0} Fib(0) = 0 \qquad Fib(n+2) = Fib(2) = Fib(1) + Fib(0)$$

$$= 1 + 0$$

$$= 1 \quad = \sum_{i=0}^{0} Fib(0).$$

True for $n = 0$.

By IH: It is true for $0 \ldots n-1$ then we need to prove for n.

$$\Rightarrow \sum_{i=0}^{n-1} Fib(i) = Fib(n+1) - 1 \qquad — (1)$$

$$\& \sum_{i=0}^{n-2} Fib(i) = Fib(n) - 1$$

Adding $Fib(n)$ to both sides in (1)

$$\Rightarrow Fib(n) + \sum_{i=0}^{n-1} Fib(i) = Fib(n+1) - 1 + Fib(n).$$

$$\Rightarrow \sum_{i=0}^{n} Fib(i) = \underbrace{Fib(n+1) + Fib(n)}_{Fib(n+2).} - 1$$

$$\Rightarrow \sum_{i=0}^{n} Fib(i) = Fib(n+2) - 1$$

Proved by induction

Practice Problems (loop Invariant)

Correctness of Merge, bubble sort, selection sort & factorial pgm.

Merge (A, p, q, r)

$n_1 = q - p + 1$

$n_2 = r - q$

let $L[1 \ldots n_1+1]$ and $R[1 \ldots n_2+1]$ be new arrays

for $i = 1$ to n_1

  $L[i] = A[p+i-1]$

for $j = 1$ to n_2

  $R[j] = A[q+j]$

$O(n/2)$.

$L[n_1+1] = \infty$

$R[n_2+1] = \infty$

$i = 1$

$j = 1$

Pre: $i = 1$, $j = 1$, $L[1 \ldots n_1+1]$, $R[1 \ldots n_2+1]$

$L[n_1+1] = \infty$ & $R[n_2+1] = \infty$

for $k = p$ to r

  if $L[i] \le R[j]$

    $A[k] = L[i]$

    $i = i + 1$

  else $A[k] = R[j]$

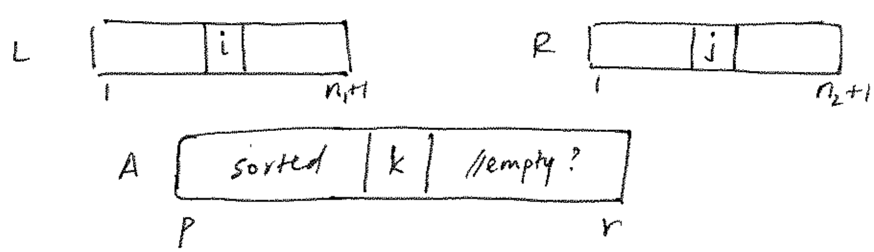
    $j = j + 1$

loop in merge $O(r - p + 1)$

$= O(n)$

Proof: Post: returns sorted array $A[p \ldots r]$

LI:



$A[p \ldots k-1]$ = sorted. $\le A[k]$

$A[k]$ = to be assigned $\min(L[i], R[j])$

$A[k+1 \ldots r]$ = not ~~filled~~ assigned.

$L[1 \ldots i-1] \le L[i] \le L[i+1 \ldots n_1+1]$

$R[1 \ldots j-1] \le R[j] \le R[j+1 \ldots n_2+1]$

Initialisation: $k = p$

$A[p \cdots p] = A[k] =$ ~~min~~ to be assigned.

$A[k+1 \cdots r] = A[p+1 \cdots r]$ not yet assigned.

Sorted Arrays

$$\left\{\begin{array}{l} L[1 \ldots i-1] = L[1..0] = \phi \le L[i] = L[1] \le L[2 \ldots n_1+1] \\ R[1 \ldots j-1] = R[1..0] = \phi \le R[j] = R[1] \le R[2 \ldots n_1+1] \end{array}\right.$$

Maintainance:

Case 1: $L[i] \le R[j]$

$\Rightarrow A[k] = L[i] = \min(L[i], R[j])$

$A[p \ldots k-1] =$ sorted $\le A[k]$ L & R are sorted

Since, $L[1 \ldots i-1] \le L[i] \le L[i+1 \ldots n_1+1]$ and everytime the ~~first~~ min
& $R[1 .. j-1] \le R[j] \le R[j+1 \ldots n_2+1]$ of two is being added to A starting from their first element.

$A[k+1 .. r] =$ not assigned.

Valid on both cases

Case 2: $L[i] > R[j]$

$\Rightarrow A[k] = R[j] = \min(L[i], R[j])$

$j{+}{+};$

Termination:

$k > r$

$\boxed{A[p \ldots r] = \text{sorted}} \le A[k] = A[r+1] = \phi$ ✓.

→ Post

$A[k] = \phi$

$A[k+1 .. r] = A[r+2 .. r] = \phi$

Bubble Sort (A) // sorts $A[1..n]$ in $\underline{O(n^2)}$

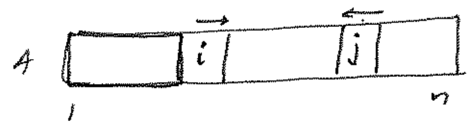
for $i = 1$ to $A.length - 1$
    for $j = A.length$ downto $i+1$
        if $A[j] < A[j-1]$
            exchange $A[j]$ with $A[j-1]$

$\left.\left.\begin{matrix} \text{for } j \ldots \\ \text{if} \ldots \\ \text{exchange} \ldots \end{matrix}\right] \text{loop2} \right] \underline{\text{loop1}}$

<u>Proof</u>: <u>~~Post~~ Pre</u>: None for loop 1, <u>loop 2</u>: $1 \leq i < n$

<u>Post</u>: loop1: $A[1 \ldots n]$ sorted. <u>loop2</u>: ~~$A[j..] < A[j+1..n]$~~ $A[i+1] < A[i+2..n]$

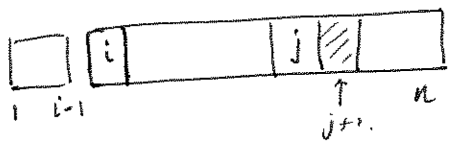
<u>LI</u>: Loop 1:



$\underline{j > i}$

$A[1 \ldots i-1]$ = sorted $< A[i] \Rightarrow \underline{A[1 \ldots i] = \text{sorted}}$.

loop 2 :



~~$A[j+1]$~~ $\Rightarrow A[j+1] > A[j]$ $\quad A[j] \leq A[j+1..n]$

<u>Loop2</u>: <u>Initialisation</u>: $A[j+1] = A[n+1] = \phi > A[j]$ ✓.

<u>Maintainance</u>: ~~$A[j] > A$~~

<u>Case1</u>: $A[j] > A[j-1]$
$j \to j-1$
no exchange, LI ~~preserved~~ persists!

<u>Case 2</u>: $A[j] < A[j-1]$
$\Rightarrow$ Exchange $A[j]$ & $A[j-1]$
$\Rightarrow$ $A[j]$ becomes $> A[j-1]$
& $j \to j-1$
LI persists!

<u>Termination</u>: <u>on</u> $j < i+1 \Rightarrow \underline{j = i}$
$\Rightarrow A[i+1] < A[i+2]$
$A[i+2] < A[i+3 \ldots n]$ from <u>LI</u>.
$\Rightarrow A[i+1] \leq A[i+2..n]$
<u>Post</u>

Loop 1: Initialisation: $A[i] = A[1]$ = ~~sorted~~ single element.

$A[1 \ldots i-1] = A[1 \ldots 0]$ = sorted ✓ $< A[i]$

$\Rightarrow A[1 \ldots i] = A[1 \ldots 1]$ = single element = sorted!

Maintainence:

loop 2 runs and posts: $A[i+1]$ & $A[i+2 \ldots n]$

$i = i+1$

$A[1 \ldots i-1]$ = sorted. $< A[i]$

$i \to i+1$. $\Rightarrow A[1 \ldots i]$ = sorted $< A[i+1]$

which is true since according to loop 2 running from $j = n$ to i, $\Rightarrow A[i] < A[i+1 \ldots n]$

$\Rightarrow A[i] <$ any value in $A[i+1 \ldots n]$

hence, the smallest value in

$A[i+1 \ldots n] > A[i]$.

$\Rightarrow A[1 \ldots i+1]$ = sorted.

Termination: $i = n$

$A[1 \ldots n]$ = sorted according to LI!

Selection sort (A)

```
for i = 1 to n
    key = A[i]
    for j = i+1 to n
        if key < A[j]
            exchange ~~key~~ A[i] and A[j]
            key = A[i];
```

loop2

loop1.

Proof : **Pre** : loop 1 : none,

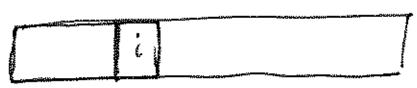
loop 2 : key = A[i]

Post : loop 1 ; A[1...n] sorted.

loop2 : A[i] = least element among A[i+1 ... n]

LT :

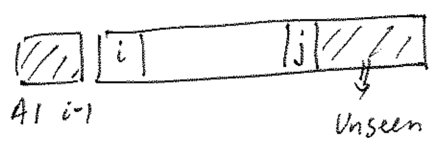
loop1 :



A[1.. i-1] = sorted

A[1] < A[2] < ... < A[i-1]

loop2 :



A[i] < ~~A[j+1...n]~~ A[i+1...j]

A[j+1...n] = unseen.

```
factorial (n)
 mul = 1
for  i = 1 to n
     mul *= i
return mul
```

Proof: Pre : $mul = 1$

Post : $mul = n!$

Loop Invariant : $mul = i! = 1 \times 1 \times 2 \times 3 \times 4 \times 5 \times \dots \times i$

Initialisation : $i = 1$

$mul = mul * 1 = 1 * 1 = 1! = i!$ ✓.

Maintainance : $1 < i < n$.

$mul = mul * i$

$mul = (1 \times 2 \times 3 \dots i-1) * i$

$= 1 \times 2 \times 3 \dots \times i-1 \times i = i!$

Termination : ~~mul~~ → $i = n+1$

$mul = n!$ Post