(20) $\lceil x + y \rceil = \lceil x \rceil + \lceil y \rceil$

$x = 1.5, \ y = 2.4$

$\lceil 1.5 + 2.4 \rceil = \lceil 3.9 \rceil = 4$

but, $\lceil 1.5 \rceil + \lceil 2.4 \rceil = 2 + 3 = 5$

hence, disproved as $4 \neq 5$.

(21) At start of each iteration of the while loop, the Subarray $A[n-1:1]$ will be converted into bits.

Initialization: If m is the integer represented by array $b[0 \dots k-1]$, then $n = t \cdot 2^k + m$.

$k = 0, \ t = n, \ m = 0$ (array is empty)

$n = n \cdot 2^0 + 0$

Maintenance: $n = t \cdot 2^k + m$ Assume true.

If t is even then: $t \bmod 2 = 0$, m unchanged

$t: t/2 \ ; \ k = k+1 \Rightarrow (t/2) \, 2^{(k+1)} + m$

$\Rightarrow t \cdot 2^k + m = n$

If t is odd, then $t \bmod 2 = 1$, $m = m + 2^k$

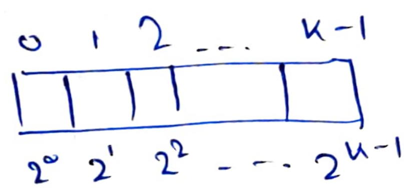
$b[k+1]$ is set to 1.

$t : (t-1)/2$

$k : k+1$

$\Rightarrow \frac{(t-1)}{2} \times 2^{(k+1)} + m + 2^k$

$\Rightarrow t \cdot 2^k + m = n.$



Termination: $t = 0$

$n = 0 \cdot 2^k + m = m$

$n = m.$ (proved)

(22) pre locateMax (A, x)

$A[n-1] \Rightarrow$ will contain the max^{m} no.

Initialization: the array contains only one element then $A[0]$ will be the max.

Maintainance: for $n = k$ Assume true, $A[k-1]$ contains largest element
Then for $A = k+1$ swap element of $(k-1)^{th}$ position to k^{th} position if $A[k-1]$ is larger else $A[k]$ is already larger. ~~else A~~
returning $A[n-1]$ which is $A[k]$.

Termination: when $i = n$ the loop terminates returning. $A[n-1]$.

(23) ~~$A[(n-1)/2 : i : A/2 + i]$~~ is the loop invariant.
$A[i : j]$.

Initialization: if the array. is of odd length,
$A[\frac{n-1}{2}] = A[\frac{n}{2}]$, so only one element present in array. so ~~reversed~~ array is reversed.
if array is of even. length $A[\frac{n}{2}]$ and $A[\frac{n-1}{2}]$ interchange thus, array is reversed.

Maitenance: for $i = k$ & $j = l$ assume true.
and. swap $A[k]$ with $A[l]$.
then for $(k-1)^{th}$ & $(l+1)^{th}$ element swap.
the ~~val~~ desired values of the array's position.
This will continue till k ~~bee~~ reaches 0 &
$l+1$ becomes $n-1$.

Termination: when $i =$ ~~n~~ 0 & $j = n-1$ array ~~sw~~ position swaping. stops. & we get the reversal of an array. of size n.
So, loop terminates.

(24)

loop invarint: $A[lo] \leq$ target $\leq A[hi]$

Initialisation: for $n = 1$, $lo = 0$ & $hi = 1-1 = 0$

thus, $mid = 0$

So target is in ~~the~~ 0th position & it will always lie in $A[0]$.

$\therefore A[lo] \leq$ target $\leq A[hi]$

$A[0] \leq$ target $\leq A[hi]$

Maintanance:

for $n = k$, $lo = 0$, $hi = k-1$ & $mid = (k-1)/2$

Now, if we find target in mid, we return it else if, target element is greater than mid element we increas $lo = mid + 1$ or else target element less than mid we decrease $hi = mid - 1$. In case there is no such case where target is found to be less than. $A[lo]$ or greater than $A[hi]$. ~~i.e. & lo & hi~~

Termination: if $hi <= lo$

loop terminates. or. mid = target. then also loop terminates.

(25)

(i) float pow (float n, int a)

x^a

Base case: if $a == 0$ return 1.0

$x^0 = 1.0$

Inductive case: for $a == k$ assume true

if $a \% 2 != 0$

$\therefore$ a is odd

so return $x \times x^{k-1}$

$\Rightarrow x^k$

if $a \% 2 == 0$

$x^{2(k/2)} \Rightarrow x^k$

for c = ...

for $a == k+1$

if $a \% 2 \,!= 0$

return $x \times x^{(k+1)-1} \Rightarrow x \times x^{k}$

$\Rightarrow \underline{x^{k+1}}$.

if $a \% 2 == 0$

return $x^{2(k+1)/2} \Rightarrow \underline{x^{k+1}}$.

Both are in the form of x^{a}.

$\therefore$ It holds true.

(ii)

(26) $f(n) = O(g(n))$ or $g(n) = O(f(n))$

(a) $f(n) = n(n-1)/2$ and $g(n) = 6n$

$f(n) = (n^2 - n)/2$ and $g(n) = 6n$

$g(n) = O(f(n))$

$0 \le 6n \le cn^2$ $\forall$ $\cancel{\quad}$ & $n \ge 1$

$c = 6$

(b) $f(n) = n + 2\sqrt{n}$ and $g(n) = n^2$

$f(n) = n + 2\sqrt{n}$ and $g(n) = n^2$

$f(n) = O(g(n))$

$0 \le n \le c \cdot n^2$ $\forall n \ge 1$ & $\cancel{\quad}$

$\underline{c = 1}$

(c) $f(n) = n + \log n$ and $g(n) = n\sqrt{n}$.

$f(n) = n + \log n$ and $g(n) = n^{3/2}$

$f(n) = O(g(n))$

$0 \le n^{\cancel{2}} \le c \cdot n^{3/2}$ $\forall n \ge 1$

$\underline{c = 1}$

(d) $f(n) = n\log n$ and $g(n) = n\sqrt{n}/2$

$f(n) = O(g(n))$

$0 \le n^{3/2} \le c.n\log n \Rightarrow 0 \le \sqrt{n} \le c\log n$.

$\forall n \ge 1$, $c = 5$

e) $f(n) = 2(\log n)^2$ and $g(n) = \log n + 1$

$g(n) = O(f(n))$

$0 \le \log n \le c.(\log n)^2$. $\forall n \ge 1$

$\underline{c = 300}$

27 (a) $2n^2+1 = O(n^2)$

True.

$0 \leq 2n^2+1 \leq c.n^2$. for $c=3$.

$\forall n \geq 1$

(b) $n^2(1+\sqrt{n}) = O(n^2)$

false.

as, $0 \leq n^2 + n^{5/2} \leq c.n^2$. is not possible

for any value c, $\forall n \geq 1$

as $n^{5/2} \geq cn^2$

(c) $n^2(1+\sqrt{n}) = O(n^2 \log n)$

true.

$0 \leq n^2 + n^{5/2} \leq c.n^2 \log n$.

as. $\sqrt{n} \leq c.\log n$. $\forall n \geq 1$

(d) $3n^2+\sqrt{n} = O(n + n\sqrt{n} + \sqrt{n})$

false.

as. $0 \leq 3n^2 \leq c.n^{3/2}$ is not possible

for any. value. c, $\forall n \geq 1$

(e) $\sqrt{n}\log n = O(n)$

True.

$0 \leq \sqrt{n}\log n \leq c.\sqrt{n}.\sqrt{n}$., $\forall n \geq 1$

$c=1$ as. $\log n = O(\sqrt{n})$

(f) $\lg n \in O(n)$

True.

$0 \leq \log n. \leq c.n.$

for $c=1$, $\forall n \geq 1$

(g) $n \in o(n\log n)$ True.

$0 \leq n \leq c \cdot n\log n$. ~~$1 = o(\log n)$~~

$c = 1$, $\forall n > 1$

(h) $n\log n \in o(n^2)$

True.

$0 \leq n\log n \leq c \cdot n^2$. , $c = 1$ & $\forall n > 1$

w.k.t. $\log n = o(n)$.

(i) $2^n \in \Omega(6^{\ln n})$

True.

$6^{\ln n} = n^{\ln 6}$.

$2^n \geq c \cdot n^{\ln 6}$, $\forall n \geq 1$ & $c = 1$.

(j) $(\log n)^3 \in o(n^{0.5})$