5.4 Recursive Algorithms

17. Describe a recursive algorithm for multiplying two nonnegative integers x and y based on the fact that $xy = 2(x \cdot (\frac{y}{2}))$ when y is even and $xy = 2(x \cdot \lfloor \frac{y}{2} \rfloor) + x$ when y is odd, together with the initial condition $xy = 0$, when $y = 0$

```
procedure multiply (x, y : nonnegative integers)
if y = 0 then return 0
else if y is even then
        return 2 · multiply (x, y/2)
else return 2 · multiply (x, (y-1)/2) + x.
```

23. Devise a recursive algorithm for computing n^2 where n is a nonnegative integer, using the fact that $(n+1)^2 = n^2 + 2n + 1$. Then prove that this algorithm is correct.

```
procedure square (n : nonnegative integer)
if n=0 then return 0
else return square(n-1) + 2(n-1) + 1
```

By mathematical induction: Let $P(n)$ be the statement that this algorithm correctly computes n^2.

Basis step: $0^2 = 0 \Rightarrow$ true.

Inductive step: Assume that algorithm works correctly for input k. Then for input $k+1$ it gives as output $k + 2(k+1-1) + 1$. By the inductive hypothesis, its output at k is k^2. at $k+1$ is $k^2 + 2(k+1-1) + 1 = k^2 + 2k + 1 =$ $= (k+1)^2$. ▨

37. Give a recursive algorithm for finding the reversal of a bit string.

procedure reverse ($b_1, b_2 b_3 \ldots b_n$: bit string)
if $n=0$ then return λ
else return b_n reverse ($b_1 b_2 \ldots b_{n-1}$)

39. Prove that the recursive algorithm for finding the reversal of a bit string is correct.

By mathematical induction:

Basis step: $n=0 \Rightarrow \lambda$ true

Inductive step: algorithm true, when $n>0$ because the reversal of a string consists of its last character followed by the reversal of its first $n-1$ charecter.

45. Use a merge sort to sort b, d, a, f, g, h, z, p, o, k into alphabetic order.

bdafg,hzpok

bda,fg hzp,ok

bda fg hzp ok

bd a f g hz p o k

b d h z

---

b d a f g h z p o k

bd fg hz ko

abd hpz

abdfg khopz

abdfghkopz

49. Prove that the merge sort algorithm is correct.

By strong induction:

Basis step: if $n=1$, algorithm does nothing, which is correct.

Inductive step: If $n=k+1$, then the list splits into 2 lists L_1, L_2. By inductive hypothesis mergesort correctly sorts each of these sublists, and so it remains only to show that mergesort correctly merges two sorted lists into one. This is clear, since with each comparison, the smallest element in $L_1 \cup L_2$ not yet put into L is put there.