Recursion

Many forms

- Ch 2.4 Recurrence relations
- Ch 5.3 Recursive definitions
- Ch 5.4 Recursive algorithms

Recurrence relations (Ch 2.5)

Recursive definitions of sequences

- E.g. 1. $a_1 = 1$, $a_n = a_{n-1} + 2$
- E.g. 2. $a_1 = 1$, $a_n = 2a_{n-1}$
- E.g. 3. $a_1 = 1$, $a_n = na_{n-1}$
- E.g. 4. $a_0 = 1$, $a_1 = 1$, $a_n = a_{n-1} + a_{n-2}$

Recursive definitions of sequences

- $S_n = S_{n-1} + n$

Recursive definitions (Ch 5.3)

Recursive definitions of sets

- E.g. 1. 3 $\in$ S, If x $\in$ S, y $\in$ S then x+y $\in$ S
- E.g. 2. 3 $\in$ S, If x $\in$ S, y $\in$ S then x*y $\in$ S
- E.g. 3. 3 $\in$ S, If y $\in$ S then 3*y $\in$ S

- E.g. 4. Definition of set of binary strings

Recursive algorithms (Ch 5.4)

Nonrecursive and Recursive definitions of n!

factorial(n) // n nonnegative integer

- If n=0 then return 1
- else return n*factorial(n-1)

Nonrecursive and Recursive definitions of a^n

power(a,n) // a nonzero real,b nonnegative integer

- If n=0 then return 1
- else return a*power(a, n-1)

Recursive algorithms (Ch 5.4)

Recursive algorithm for Fibonacci numbers.

fib(n) // n nonnegative integer

- If n=0 then return 0
- else if n= 1 then return 1
- else return fib(n-1)+fib(n-2)

Recursive algorithms (Ch 5.4)

GCD computation

gcd(a,b) // a,b nonnegative integers, a < b

- If b=0 then return a
- else return gcd(b, a mod b)

Merge Sort

Next: Recursive algorithm analysis

Solving recurrence relations

Run-time analysis of recursive algorithms