21p [1,2,3]['a',5', c'] Lecture 4. = [(1, |a|), (2, |b|), (3, |c|)]21p :: [a] → [b] → [(a,b)] zip [] ys = [] Zip XS [] = [] Zip (x:xs) (y:ys) = (x,y): zip xs ys Zip [1,2,3,4,5] [a,16,10] = (1, 'a'): Zip [2,3,45] ['b', 'c'] (1, 'a'): (2, 'b'): Zip [3,4,5] ['c'] ziplith (add) [1,2,3] [4,5,6] = [()+4), (2+5), (3+6) add 14 add 25 add 36 ziplith :: (a>b>c) > [a] → [b] → [c] ZipWith of [] ys = [] ZipVith of xs () = [] ZipWith f (X:XS) (Y:YS) = f x y : zipWith f xr ys [c] [a] [b] map :: (a+b) -> [a] -> [b]

fib ::
$$|n+ \rightarrow |n+ |$$

fib $0 = 1$

fib $1 = 1$

fib $n = f_{b}(n-1) + f_{ib}(n-2)$,

fib (10)

fibs: $|n+1| + |n+1|$

fibs: $|n+1| + |n+1|$

fib $n = f_{ib}s | |n$

(+) :: $|n+-1| + |n+1|$

add $= (+)$

Higher-order functions. A higher-order function takes a function as a parameter. map:: (a > b) > [a) > [b] Zidlike: (a-b+c) - [a7 -) [b] - [c] filter : (a -> Book) -> [a] -> [a] [1, 2, 3, 4] 1+(2+(3+(4+0))) sum xs = foldr (+) 0 xs : Int [1, 2, 3, 4] = 1:2:3:4: L7 7 7 7 7 1* 2* 3*4 * 1 fac n = fadr (x) 1 [1.. n] folar :: (a > b + b) + b + [a] + b fddr f k [] = k foor f k (x:xs) = f x (folder f k xs) $f:: a \rightarrow b \rightarrow b$. $(+):: lnt \rightarrow (nt \rightarrow lnt)$ $feo:: Char \rightarrow lnt \rightarrow lnt$ a:= Char $bar:: lnt \rightarrow Char \rightarrow lnt$ b:= lnt