## REKORSHION

maximum': (ord a) => [a] -> a maximum' [] = error " maximum of empty list" maximum [x] = 2

maximum (x: 213)

2 > max Tail = 2

otherwise = max Tail where maxTail = maximum' as

maximum': (orda) => [a] -> a

maximum' [] = error "maximum of empty list" maximum' [x] = x

maximum' (x:xs) = max x (maximum' xs)

addition + compansion replicate! :: (Numi, Ordi) => i → a → [a]

replicate! nx

 $1 n \leftarrow 0 = []$ 

! otherwise = x: replicate! (n-1) x

take :: (Numi, ordi) => i -> [a] -> [a] take n -

| n <= 0 = to[] (000) p = 100 move

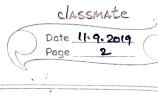
take! - [] = []

take n (x:xs) = x: take (n-1) 2s

reverse' :: [a] -> [a]

reverse = []

YEVERSE! (X:XS) = XS+4-ENJ reverse! XS +4 [x]



repeat : a -> [a] repeat x = x: repeat x

 $Zip! : [a] \rightarrow [b] \rightarrow [(a,b)]$ zip' [ [ ] = []

zip' [] - = []

elem! \_ [] = False

| a == x = True.

1 otherwise = a element

quicksort: (ord a) => [a] -> [a]

let smaller sorted = quicksort [a | a < xs, a <= xc]

in smallersorted ++ [x] ++ [biggersorted

2 quicksort [10, 2, 8, 3, 1, 6, 7, 4, 2, 3, 4, 8, 9]

[1,2,2, 3,13,14,4,5,16,17,8,93,10] dissipant to

2 quicksort "the quick brown fox jumps over the lazy day"

Compare With Hundred .: (Num a Ord a)

Company tricks Hundred 31 : Company

abadeee fighhijklimn oo oopgrest tuu vwxyz' 131

bigger sorted = quicksort [a lak-xs, axx]

elem' a (x:xs)

quicksort [] =

quicksort (x:xs) =

zip' (x:25) (g:ys) = (x,y): zip' 25 ys

elem' : (Eq a) => (a -> [a] -> Book