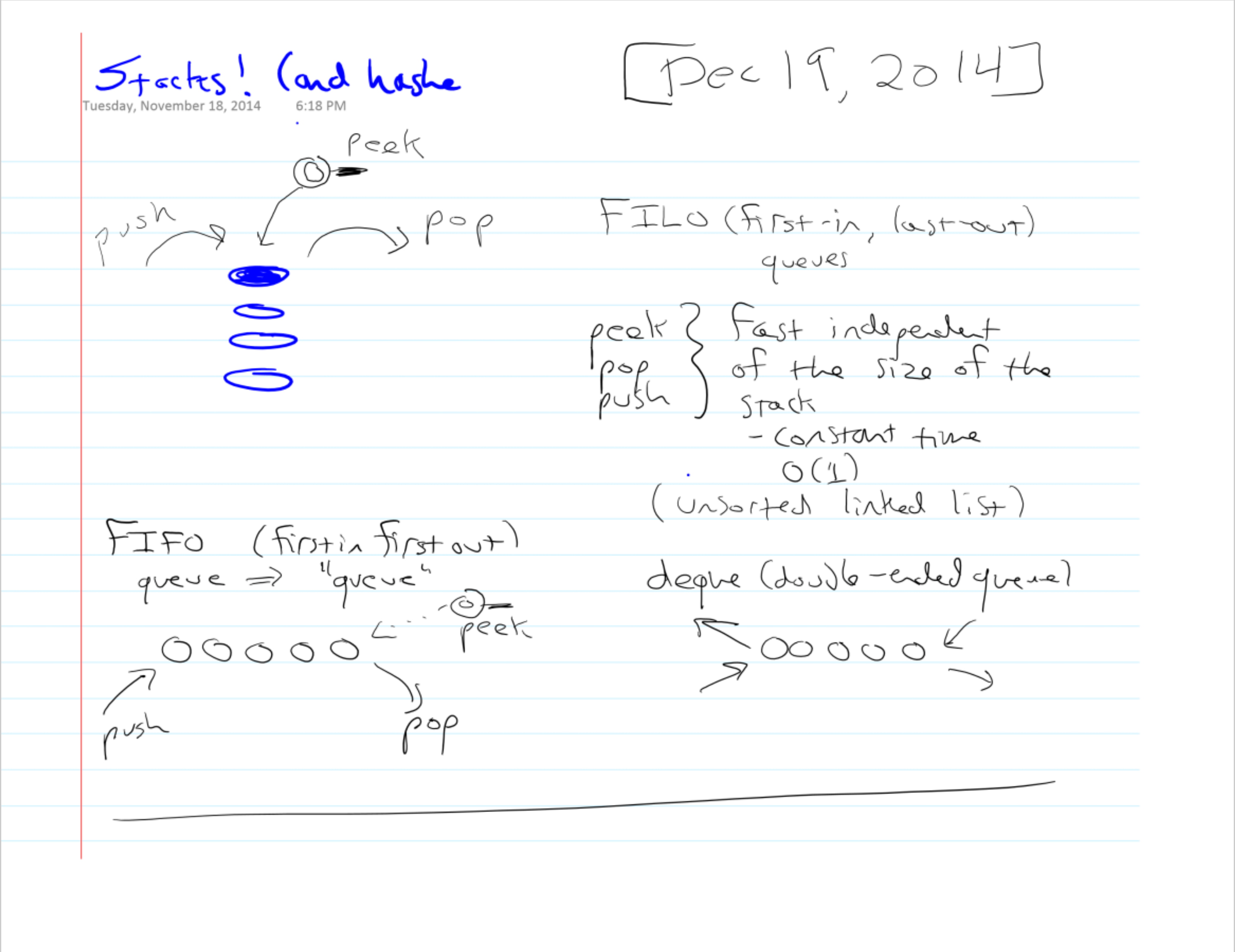
Printout

Wednesday, November 19, 2014 3:08 PM



Stacks! (and hashe

Tuesday, November 18, 2014 6:18 PM

[Dec 19, 2014]

FILO (first-in, last-out)

queues

peek, pop, push } Fast independent of the size of the stack

- constant time

O(1)

(unsorted linked list)

FIFO (first in first out)

queue => "queue"

deque (double-ended queue)

Page 2

Wednesday, November 19, 2014 3:08 PM

```
library(rstackdeque)

s <- rStack()
s <- insert_top(s, "A")
s <- insert_top(s, "B")
s <- ins .... (s, "C")
print(s)
      ↳            C
                   B
                   A

print(peek_top(s))              # print "C")

s <- without_top(s)
print(s)
                 B
                 A

g  <-  insert_top(s, "X")
```

```
g: X     s: B
   B        A
   A
```

Page 3

Wednesday, November 19, 2014 3:08 PM

```
glist  <-  as.list(g)     # "X" "B" "A" as list
```

```
rows  <-  rstack()
rows  <-  insert_top(rows, list(name = "bob", age = 24))
rows  <-  insert_top(rows, list(name = "joe", age = 27))

rowsdf  <-  as.data.frame(rows)
print(rowsdf)
```

↳

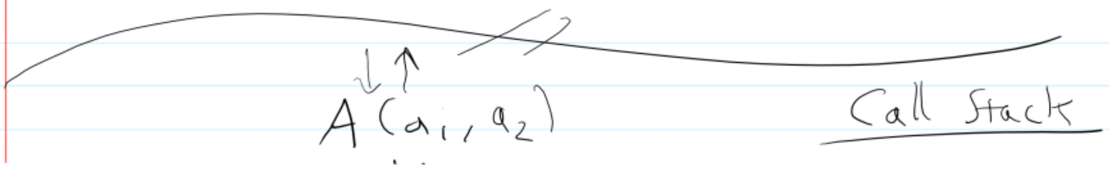
```
name   age
Bob    24
joe    72
```

```
fib  <-  Function(n) {
    CALL_COUNTER <<- CALL_COUNTER + 1
    ⋮
}
```

Page 4

Wednesday, November 19, 2014 3:08 PM

```
library(rstackdeque)

info <- rstack()

for(i in seq(1,15)) {
    CALL_COUNTER <<- 0
    row <- list(n = i, fibn = fib(i), calls = CALL_COUNTER)
    info <- insert_top(info, row)
}

library(ggplot2)

infodf <- as.data.frame(info)
p <- ggplot(infodf) +
         geom_line(aes(x=n, y=fibn))+
         geom_line(aes(x=n, y= calls))

plot(p)
```



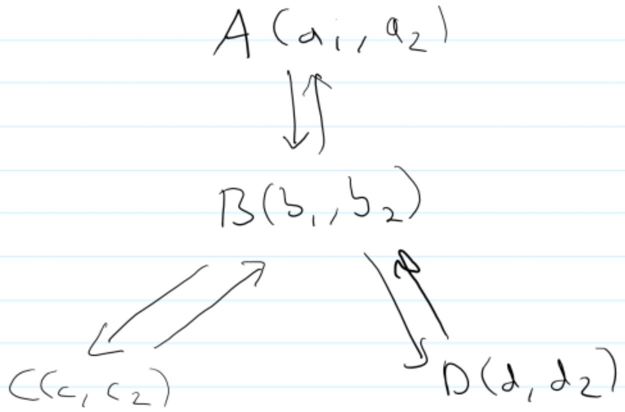
Page 5

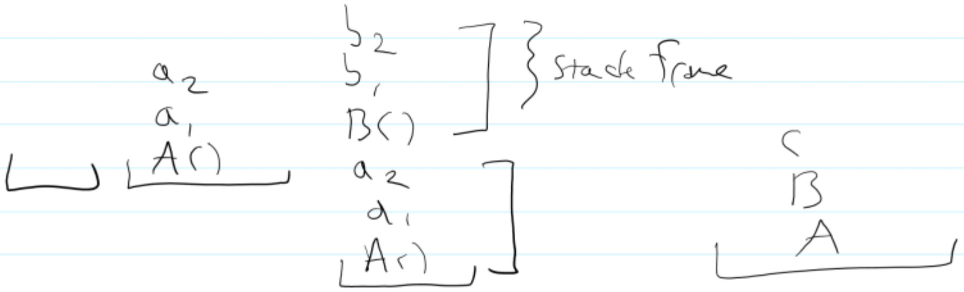
Wednesday, November 19, 2014 3:08 PM

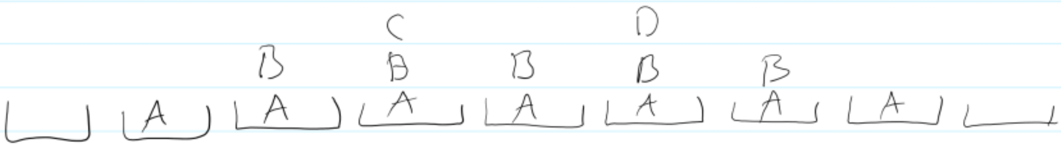
Call Stack

Every time a function is called, some info is put on this call stack.

-when it returns, the call stack pops

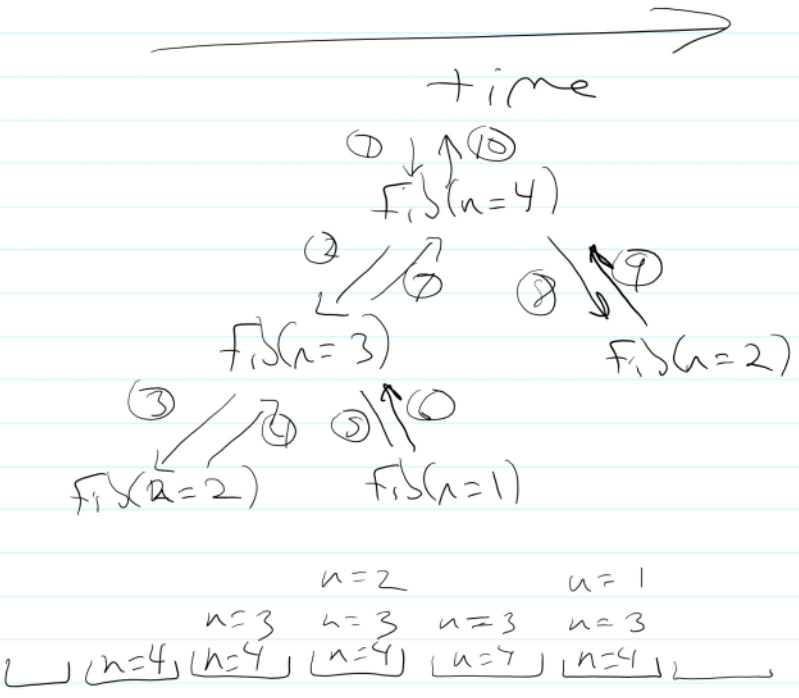






Page 6

Wednesday, November 19, 2014 3:08 PM



'Stack overflow'

```
fib <- function(n) {
    CALL_COUNTER <<- CALL-COUNTER +1
    thiscall <- str_c("fib:n=", as.character(n))
```

Wednesday, November 19, 2014 3:08 PM

```
thiscall <- str_c("fib:n=", as.character(n))
```

4

⇓

"4"

"fib:n=4"

```
CALL_STACK <<- insert_top(CALL_STACK, thiscall)
~~print(s)~~ print_string_stack(CALL_STACK)
```

Remove before every return:

```
CALL_STACK <<- without_top(CALL_STACK)
return( ... )
```

```
print ~~vecto~~ string_stack <- function(~~a~~ s) {
```

Page 8

Wednesday, November 19, 2014 3:08 PM

```
print_~~vecto~~string_stack <- function(~~x~~s) {
    print(s)
}
```

$$\frac{fib(n)}{fib(n-1)} \xrightarrow[n\to\infty]{} 1.618\ldots \quad \varphi$$ golden ratio

$$fib(n) \approx 1.618 \cdot fib(n-1)$$