Storage organization

Procedure
- declaration that associates an identifier with a statement

subdivision of run-time memory

compile time → code
             ↘ static data   → memory locations for code

run time → stack
            ↓
          free memory
            ↑
          heap - ex malloc

Activation tree
Root - main
functions are children

?3

S

Run time

enter main ()
  enter readarray()
  leave readArray ()
  enter qsort (1,9)
    enter part (1,9)
    leave part (1,9)
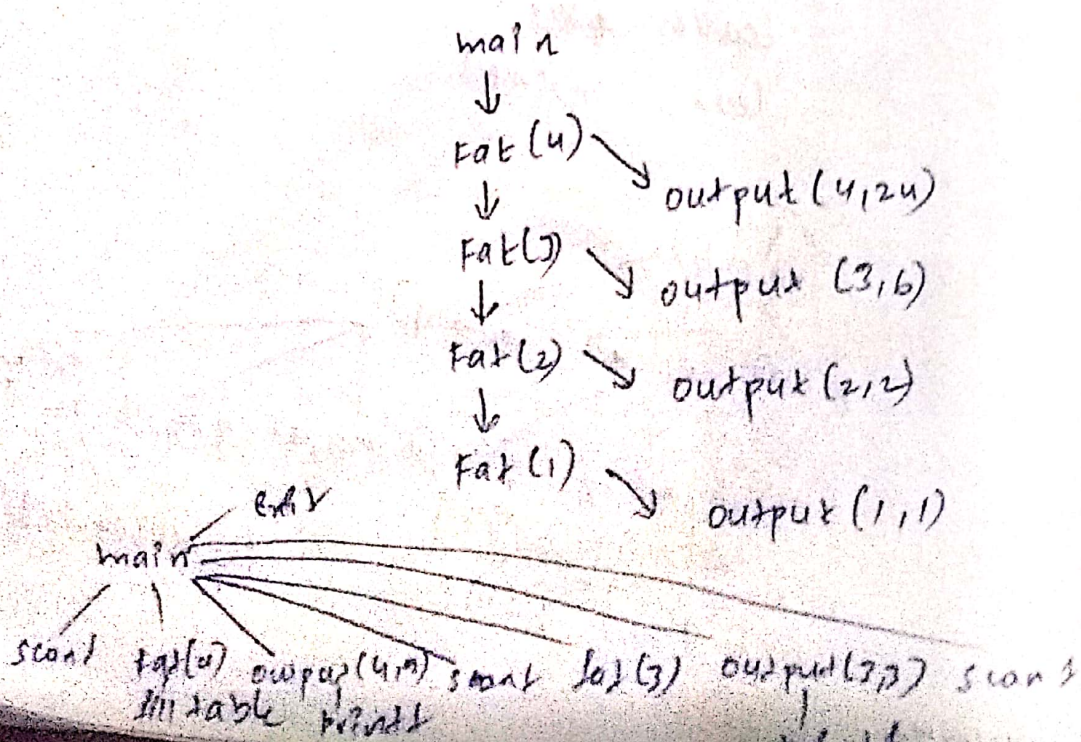   ⋮
  leave qsort (1,9)
leave main()

V.varma.

Example - Quick sort Activation tree
↳ helps us to know
all function are active
and also the control
flow

```
void output (int n, int x){
    printf(" the value of % d!, is % d \.",n,x)
}

int fat (int n){
    int x;
    if (n>1)
        x= n* Fat (n-1)
    else
        x=1
    Output (n,x)
    return x;
}

void main(){
    Fat (4);
}
```

Activation tree fat(4)

main
↓
Fat (4) ⟶ output (4,24)
↓
Fat(3) ⟶ output (3,6)
↓
Fat(2) ⟶ output (2,2)
↓
Fat (1) ⟶ output (1,1)

entr

main

scant fat(4) owpay(4,n) scant fo)(3) output(3,3) scant
fill table print

control Stack

→ keep track of live procedure activation
→ flow of control in program corresponds to
   depth first traversal of Act tree

   push - activation begins
   pop - when act. ends

Activation record

" control link - pointer
that points to activation
record of caller

access link - points to
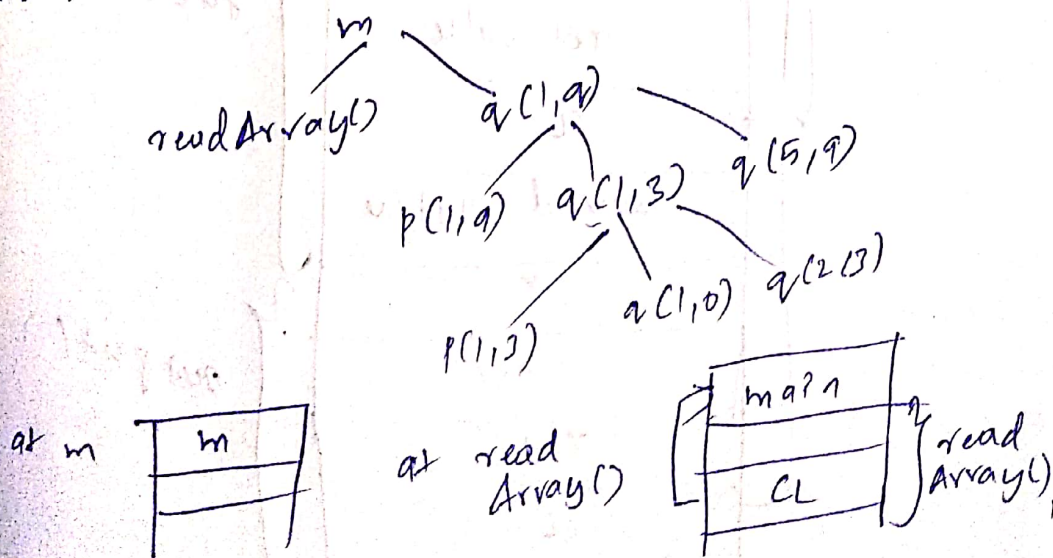non local data (maybe in
other act record)

SMS - holds into about machine
       status before procedure call

local data temporaile - temporary
value used in evaluation

Actual parameters
return value
control link
Access link
Saved machine state
local data
temporaries

returned
value - value
to be
returned

Activation record - Eg Quicksort

m
readArray()   a(1,9)
                    /   \
        p(1,9) q(1,3)   q(5,9)
              /   \ \
          p(1,3) a(1,0) q(2,3)

at m   [m]   at read   [main]   read
        [   ]   Array()   [CL]   Array(),

activation record eg.

		} main
n=4		a
ret value		fact(4)
CL		
saved status		
x		
n=3		
ret value		fact(3)
CL		
saved status		
x		
n=2		
ret value		fact(2)
CL		
saved status		
X		
n=1		fact(1)
ret value		
CL		
saved status		
v		
n=1		output()
ret		
CL		
saved status		

sp →