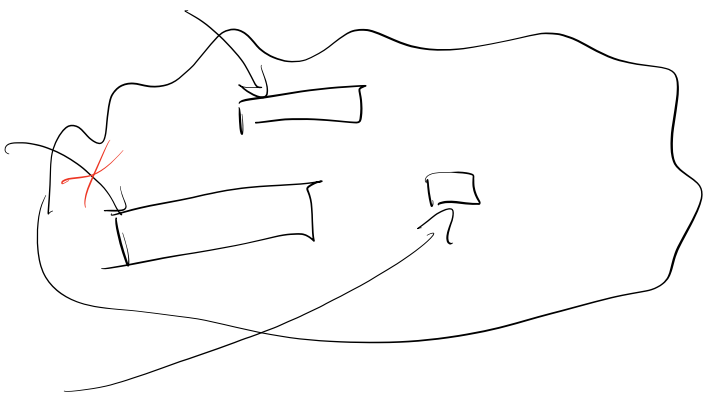
ORGANIZZAZIONE DELLA MEMORIA

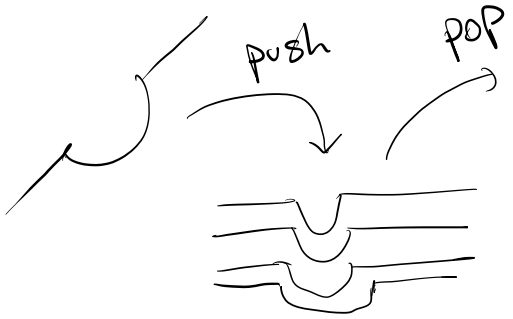
1) STACK DI ESECUZIONE
2) HEAP
3) MEMORIA STATICA (var. globali)

HEAP



new / malloc

STACK



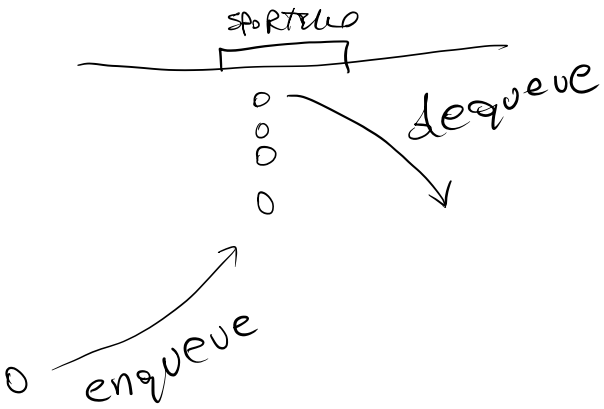
Agostino

Bianca

↑

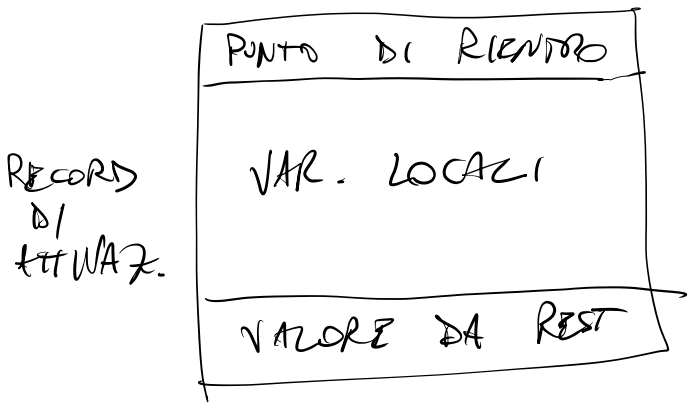
STACK

LIFO = Last-In First-Out



FIFO

RECORD DI ATTIVAZIONE



RECORD DI ATTIVAZ.

```
func f(a,b int) (c int){
    var x,y float64
    ...
}
```

```
1  func main() {
2      var x, y, ris int
3      fmt.Scan(&x)
4      fmt.Scan(&y)
5      ris = f(x, y)
6      fmt.Println(ris)
7  }
8  func f(a, b int) (c int) {
9      var x, y int
10     x = sqr(a)
11     y = sqr(b)
12     c = x + y + 1
13     return
14 }
15 func sqr(x int) (a int) {
16     a = x * x
17     return
18 }
```

59

9

49

RICORSIONE

(*) $fatt(n) \triangleq 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$

$fatt : \mathbb{N} \to \mathbb{N}$

1, 2, 3, 5, ...

CASO BASE

(**) $$fatt(n) \triangleq \begin{cases} 1 & \text{se } n = 0 \\ n \cdot fatt(n-1) & \text{se } n > 0 \end{cases}$$

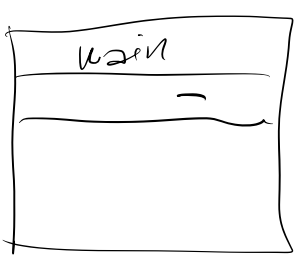
$$fitt(n) = \begin{cases} 1 & \text{se } n \geq 0 \\ 2 & \text{se } n = 1 \\ 3 * fitt(n+1) & \text{altr.} \end{cases}$$

$$fib(n) \triangleq \begin{cases} 1 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ fib(n-1) + fib(n-2) & \text{altrimenti.} \end{cases}$$

```
1  func fatt(n int)(ris int){
2      if n==0 {
3          ris=1
4      } else {
5          v := fatt(n-1)
6          ris = n*v
7      }
8      return
9  }

10 func main(){
11     fmt.Println(fatt(4))
12 }
```

24



```
func fatt (n int) int {
    if n == 0 {
        return 1
    } else {
        return n * fatt(n-1)
    }
}
```

$$fatt(n) \triangleq \begin{cases} 1 & \text{se } n=0 \\ n \cdot fatt(n-1) & \text{altr.} \end{cases}$$