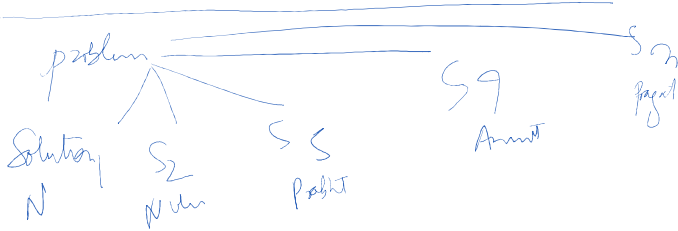
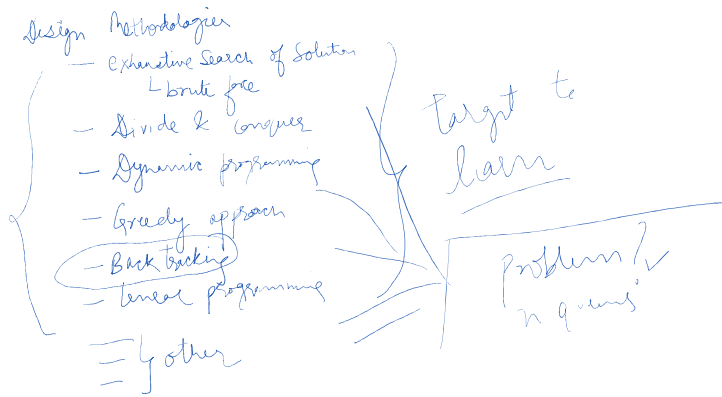


Design & Analysis of algorithm  
 what is algorithm  
 Blue print  
 Result of your thought process



Which solution is better??

Analysis of algorithms

- Comp main
- ✓ Time Complexity
  - ✓ Space Complexity  $\rightarrow$  variable
  - ✓ Implementation Complexity
  - ✓ portability  $\rightarrow$  machine  $\rightarrow$  2 machine  $\rightarrow$  acc

time required by the algorithm to solve the problem  $\rightarrow$

main()

```

{
    t1 = time()
    // ...
    t2 = time()
    return()
}
    
```

machine dependent complexity

time  $t_2 - t_1$




Dependent  
Complexity }  
return()

(72-11)

Why we are interested  
in theoretical analysis?

$M_1$   
Bad



$10^2$  ins/Sec


Student 1:  $n \lg n$

$1000 \lg 1000$  sec

$= 100$

30 sec

$M_2$   
Good



$10^3$  ins/Sec

Student 2:  $n^2$

$\frac{1000 \times 1000}{1000}$  sec

1000

716

m/c independent  $\leftarrow$   $< 1$  hr

import  
of theory

Best case ✓  
Worst case ✓  
Average case ✓

Two

1  
B

Order of  
growth

n = 10

$$\sum_{i=1}^n (n-i+1)$$

Bubble\_sort()

```

for i = 1 to length(A)-1
  for j = length(A) down to i+1
    if a[j] < a[j-1]
      a[j] <- a[j-1]
      a[j-1] <- a[j]
    end if
  end for
end for
  
```

count swaps → worst case →  $\sum_{i=1}^n (n-i)$   
 Avg case →  $\frac{n(n-1)}{2}$

$$T_{wc} = c_1(n+1) + c_2 \sum_{i=1}^n (n-i+1) + c_3 \sum_{i=1}^n n-i$$

$$= \frac{c_1 + c_2}{2} n^2 + \frac{c_1 + 3c_2}{2} n + c_1$$

Asymptotically =  $\boxed{an^2 + bn + k}$

$\sum_{i=1}^n n-i$   
 $\frac{n(n-1)}{2}$

