

# Another look at optimization problems

- 1) Optimal substructure
- 2) Overlapping subproblems

# Fibonacci substructure

```
def f(n):  
    if n <= 1: return 1  
    else: return f(n-1) + f(n-2)
```



## Number of calls to $f(n)$ ?

Let  $T(n)$  = # of calls to compute  $f(n)$