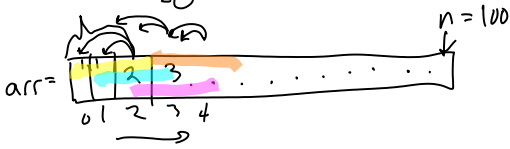
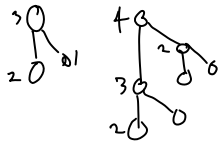


Dyn prog 1
Thursday, March 5, 2020 3:35 PM

$$fib(n) = \begin{cases} 1 & \text{if } n=0 \\ 1 & \text{if } n=1 \\ fib(n-1) + fib(n-2) & \end{cases}$$



```

fib(n):
  if n < 2: return 1
  arr := [0]^n // arr[i] = fib(i)
  arr[0] := 1
  arr[1] := 1
  
```

```

for i = 2 upto n:
  arr[i] := arr[i-1] + arr[i-2]
return arr[n-1]
  
```

$T(n) = O(n)$ $S(n) = n$

$S(n) = O(1)$

```

Fib Better (n):
  pprev := 1
  prev2 := 1
  if n < 2: return 1
  for i = 2 to n-1:
    p := prev + prev2 // p = fib(i)
    // prev = fib(i-1)
    // prev2 = fib(i-2)
    prev2 := prev
    prev := p
  return p
  
```

String is in L^*
 we have $S \cap L: \Sigma^* \rightarrow \{0,1\}$ in $O(1)$
 $S \cap L \cap W \iff W \in L$

ex. L is the set of English words

a...
 aärdark.
 ; "This is a sentence"

$$S \cap L^*(W) = \begin{cases} T & W = \epsilon \\ T & W \in L \\ \exists u, v, \text{ st. } W = uv \text{ and } u \in L_n, v \in L_n \end{cases}$$

$SinL^*(v)$

```

def  $SinL^*(w)$ :  $n = |w|$ 
  if  $w$  is empty: T
  if  $SinL(w)$ : T
  for  $i = 1$  to  $n$ :
     $\underbrace{\text{prefix}} \quad \underbrace{\text{suffix}}$ 
    let  $u, v = w[:i], w[i:]$ 
    if  $SinL(u)$  and  $SinL^*(v)$  then T
  else F // make it through loop.

```



percent:

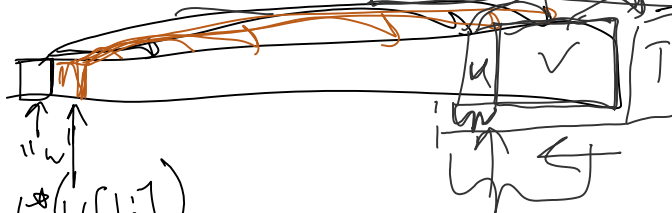
def $SinL^*(w)$:

// w is fixed

$n = |w|$

declare array

$ISL := [F]$ $ISL[i] = SinL^*(w[i:])$



$ISL[n] := T$

for $i = n-1$ down to 0:

for $j = i+1$ to n :

$u := w[i:j]$

$v := w[j:n]$

if $SinL(u)$ and $ISL[j]$ then

$ISL[i] := True$

Subset Sum:

e.g. $\downarrow, 5, 5, 10, 20$

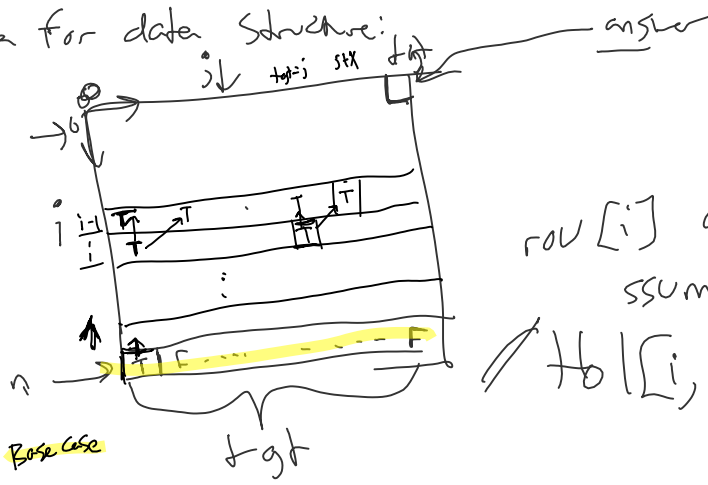
target = 25

$SSum(arr, target)$:

- T $arr = [], target = 0$

$\{ F \text{ arr} = [], \text{tgt} \neq 0$
 $(\text{ssum}(\text{rest}, \text{tgt} - x))$ $x \in \text{rest} = \text{arr}$
 OR $(\text{ssum}(\text{rest}, \text{tgt}))$

Idea for data structure:



$\text{row}[i]$ corresponds to
 $\text{ssum}(\text{arr}[i:], _)$

$\text{tbl}[i, j] = \text{ssum}(\text{arr}[i:], j)$