

CSE 20 11-2-21

1

-
- quiz today
 - lab 4 due wed.
 - Pa 5 due fri.

Review Problems on lists:

① write a fn that concatenates two lists.

```
def concat(A, B):  
    return A + B  
# end concat()
```

do it without the builtin + op.

```
def concat(A, B):
```

```
    C = list(A) # also A[:]
```

```
    for i in range(len(B)):
```

```
        C.append(B[i])
```

```
    # end for
```

```
    return C
```

```
# end concat()
```

② interleave two lists (of same len.)

same len $\begin{matrix} [1, 2, 3] \\ [4, 5, 6] \end{matrix} \} \rightarrow [1, 4, 2, 5, 3, 6]$

diff len $\begin{matrix} [1, 2, 3, 4, 5] \\ [6, 7, 8] \end{matrix} \} \rightarrow [1, 6, 2, 7, 3, 8, 4, 5]$

```
def interleave(A, B):
```

```
    C = []
```

```
    for i in range(len(A)):
```

```
        C.append(A[i])
```

```
        C.append(B[i])
```

```
    #end for
```

```
    return C
```

```
#end interleave()
```

more exercise

- add two lists (of ints)
- multiply two lists (Hadamard Product)

$$\begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \end{bmatrix} \rightarrow [4, 10, 18]$$

• scalar multiply number. list

$$3 \cdot [1, 2, 3] \rightarrow [3, 6, 9]$$

• dot product of two lists

$$\begin{bmatrix} 1, 2, 3 \\ 4, 5, 6 \end{bmatrix} \rightarrow 32$$

Ex. (helpful on Pass)

$$\text{let } L = [1, 2, 3, 4, 5]$$

create a new list without 3 (index=2)

$$M = L[:2] + L[3:]$$

now leave out index i ($0 \leq i \leq 4$)

$$M = L[:i] + L[i+1:]$$

Chap. 11: Dictionaries

A Dictionary is similar to a list.

$L = []$ or $L = \text{list}()$ # empty list

$D = \{ \}$ or $D = \text{dict}()$ # empty dictionary

Both are containers of objects,
any objects.

Difference:

in a list, indices have to be ints

in a dictionary, indices can be any
immutable object. we don't

call them indices anymore. we
call them keys

Ex.

D = dict()

D['foo'] = 'bar'

↑ ↑
key value

D['happy'] = 6

D[7] = 'blah'

D[(1, 2)] = list()

Print(D)

Both are mappings

$L = [1, 2, 3]$

<u>index</u>		<u>value</u>
0	→	1
1	→	2
2	→	3

$D = \{ 'uno': 'one', 'dos': 'two', 'tres': 'three' \}$

<u>key</u>		<u>value</u>
uno	→	one
dos	→	two
tres	→	three

```

for key in D:
    Print(key, D[key])
end for

```

Ex. Dictionary Functions. Py

Ex. Count Characters. Py

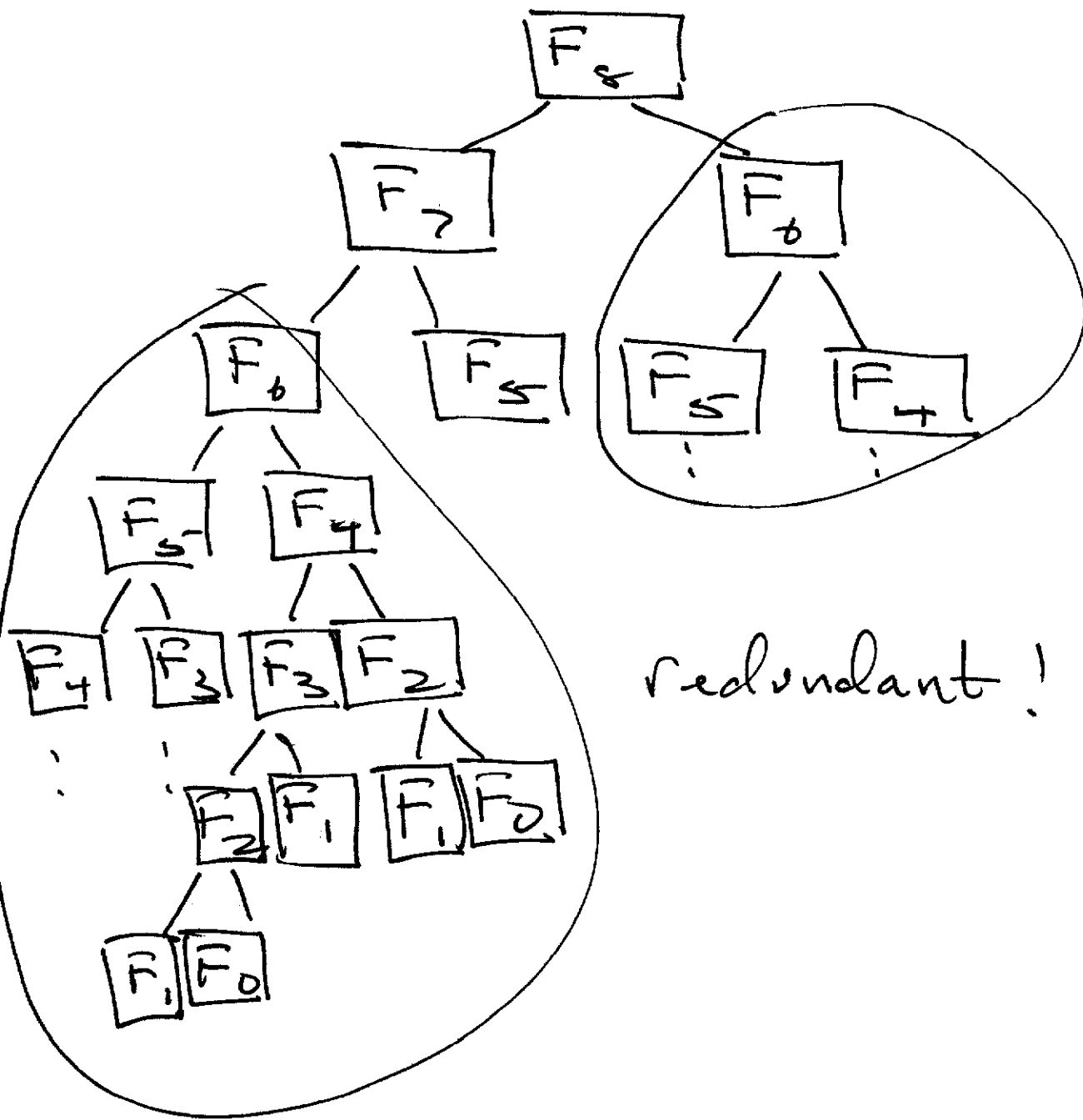
Ex. Fibonacci 3. Py
4

Fibonacci Seq:

0 1 1 2 3 5 8
└──
initial

$$F_n = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ F_{n-1} + F_{n-2} & n \geq 2 \end{cases}$$

Recursive version !



Read : chap 12 : Tuple.