Lecture 11:

- performance analysis
- problem reduction
- recursion
- [divide & conquer]

RL $\subseteq$ CFL $\subseteq$ CSL $\subseteq$ URG Top

D/NFA $\subseteq$ NPDA $\subseteq$ LBA $\subseteq$ TM

λ- calculus

C, Python, assembly

RAM model — TM

- arrays
- asymptotic behavior
- addition, subtraction, comparison, array lookup → constant time

↓ RAM model

L_1 for L_2 L_3

$T(\cancel{L_1}) + n(T(L_2) + \cancel{1(L_3)}) \in O(n)$

Reducing a problem to another

$B \longmapsto A$

$A : f : \Sigma^* \to \Sigma^*$

Problem Domain $D \subseteq \Sigma^*$

add : D pairs of integers
R integers

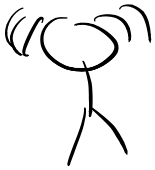
L Mem: D strings
R true/false

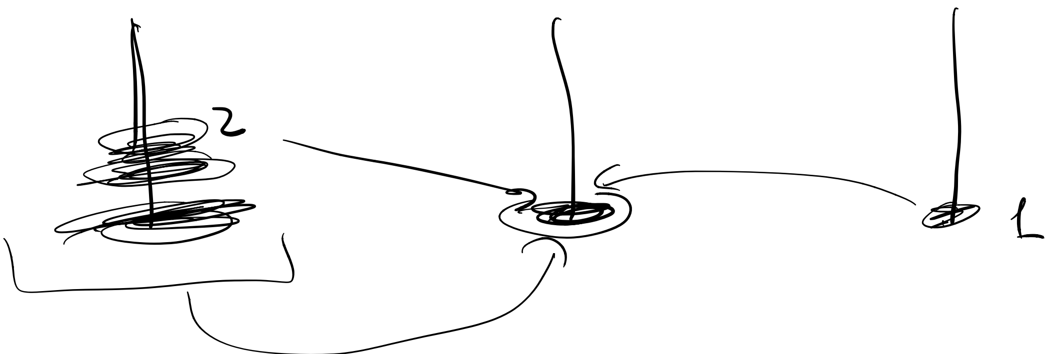
B: (x)
y = transform(x) from domain B to domain A
return A(y)

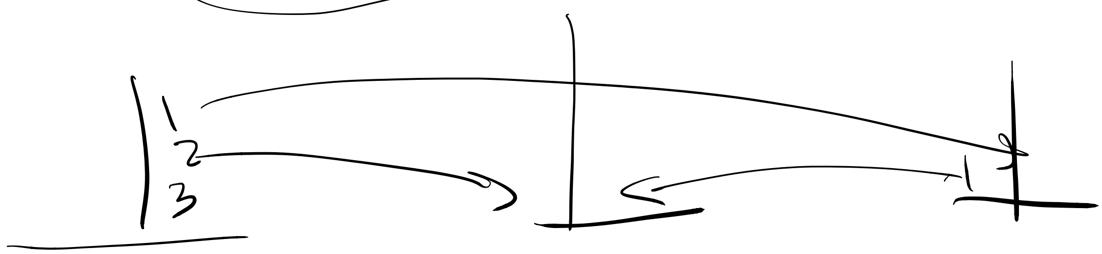
Sep30

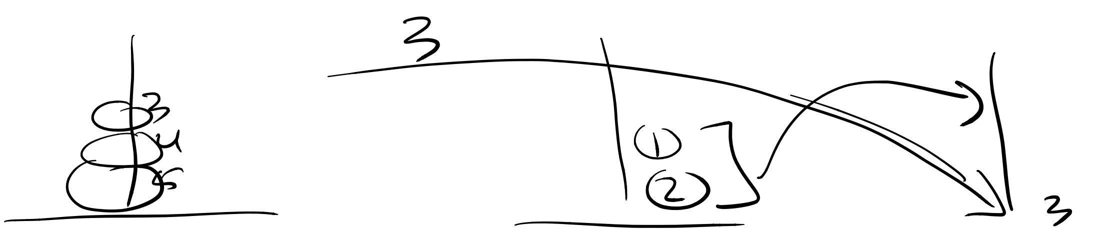
A^+

Nous









Hanoi(n, s, t, x)

n = 5; s = 1 (pole#); t = 2 (pole#); x = 3 (other pole#)

```
// move n disks from
    s to t leaving x empty (@ end)
if (n > 0):
    Hanoi(n-1, s, x, t)
t → Move disk n from s to t
    Hanoi(n-1, x, t, s)
else:
    pass
```

$H(n) \rightarrow$ # of 1-disk moves that Hanoi(n) does

$$H(0) = 0$$

$$H(k) = H(k-1) + 1 + H(k-1)$$

$$= 1 + 2H(k-1)$$

H(k) — 1

H(k-1) H(k-1) — 2

k levels

$H(k-2)\ H(k-2)\ \ldots$ 4

$\vdots$

$k-i$ 2^i moves

$$\sum_{i=0}^{k-1} 2^i = 2^k - 1$$