$F = \{f_1, f_2, \dots, f_n\}$

$H_1 = \{h_1^1, h_2^1, h_3^1, \dots, h_n^1\}$

$H_2 = \{h_1^2, h_2^2, \dots, h_n^2\}$

initial call

cookTime$(\phi, \phi, F, 0, 0)$

→ CookTime(L_1, L_2, FC, t_1, t_2)

- L_1: items in H_1 chosen for cooking
- L_2: items to be cooked in H_2 ~~already~~ already chosen
- FC: set of food items yet to be chosen for cooking
- t_1: total cooking time already allocated to H_1 (derived from L_1)
- t_2: total cooking time already allocated to H_2 (derived from L_2)

BASE :- If $FC = \phi$ return $\max(t_1, t_2)$

Recursive :- choose $f_i \in FC$.

→ $y_1 = \text{cookTime}(L_1 + f_i, L_2, FC - f_i, t_1 + h_i^1, t_2)$

→ $y_2 = \text{CookTime}(L_1, L_2 + f_i, FC - f_i, t_1, t_2 + h_i^2)$

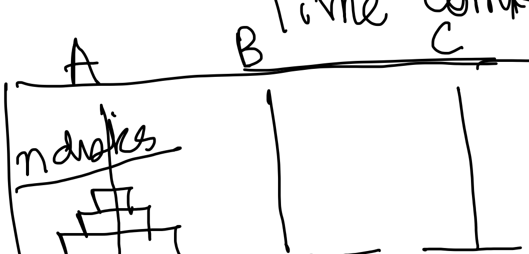
Recomposition :- $y = \min(y_1, y_2)$

return (y)

Hourglass

Proof of Correctness : Induction Principle

Time Complexity Analysis



3-peg, n-disk ToH

TOH(from, to, via, n)

Base: $n=1$

$L = \{\text{move}(\text{from}, \text{to})\}$

return (L)

recursive

$L_1 = \text{TOH}(\text{from}, \text{via}, \text{to}, n-1)$

$L_2 = \{\text{move}(\text{from}, \text{to})\}$

$L_3 = \text{TOH}(\text{via}, \text{to}, \text{from}, n-1)$

$L = L_1 \| L_2 \| L_3$

return(L)

$T(1) = 1$

$T(n) = 2T(n-1) + 1$

$= \cancel{2^{n+1} - 1}$ ✗

$\cancel{2^{n-1} + 1}$ | $2^n - 1$

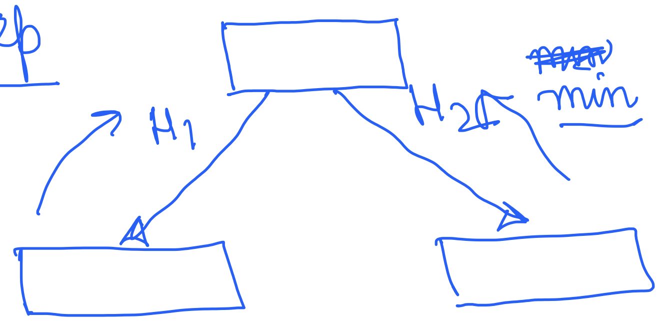
$T(n) = 2T(n-1) + 1 \quad = 2^n - 1$

Tower of Brahma

PRALAY 3 peg, 64-disk problem

↳ Time of 1 move is equivalent to ONE DAY $(2^{64} - 1)^{th}$ day

Next step



— IDENTICAL SUB-PROBLEMS ???

↳ what is the check

— FC is the same

$S_1 = L_1^1, L_2^1, FC^1, t_1^1, t_2^1$

$S_2 = L_1^2, L_2^2, FC^2, t_1^2, t_2^2$

$FC^1 = FC^2$

$t_1^1 = t_1^2$

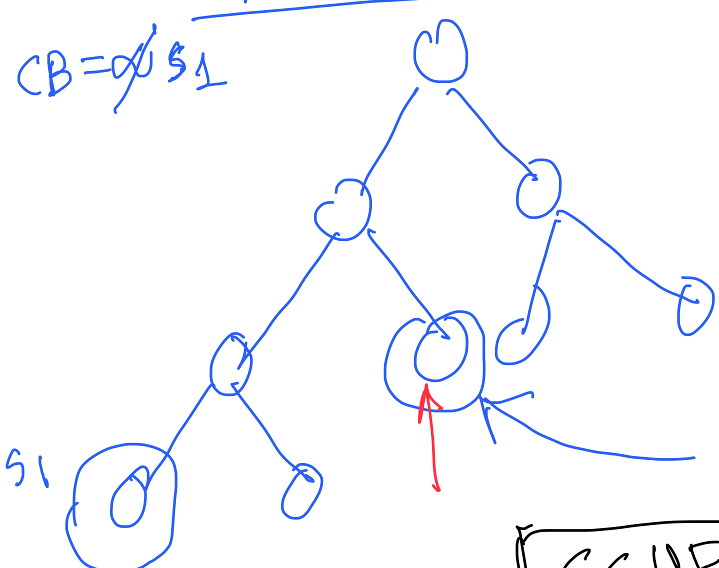
$t_2^1 = t_2^2$

— ARE THERE CYCLES? X

— CAN WE MAKE A GREEDY CHOICE BETWEEN H_1 & H_2?

Depth-first manner

$CB = \cancel{\infty}\ S_1$

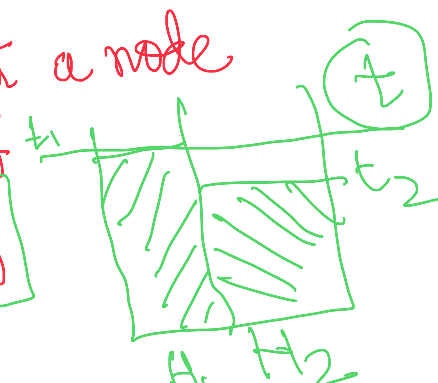


PRUNING

$t = \max\{t_1, t_2\}$ at a node

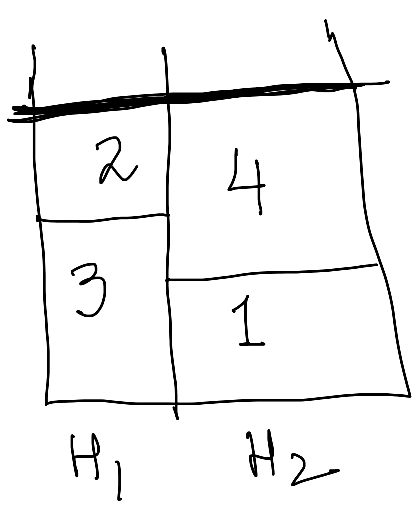
we will prune if $t \geq CB$

Fc



processors

SCHEDULING of independent TASKS



Set Partitioning