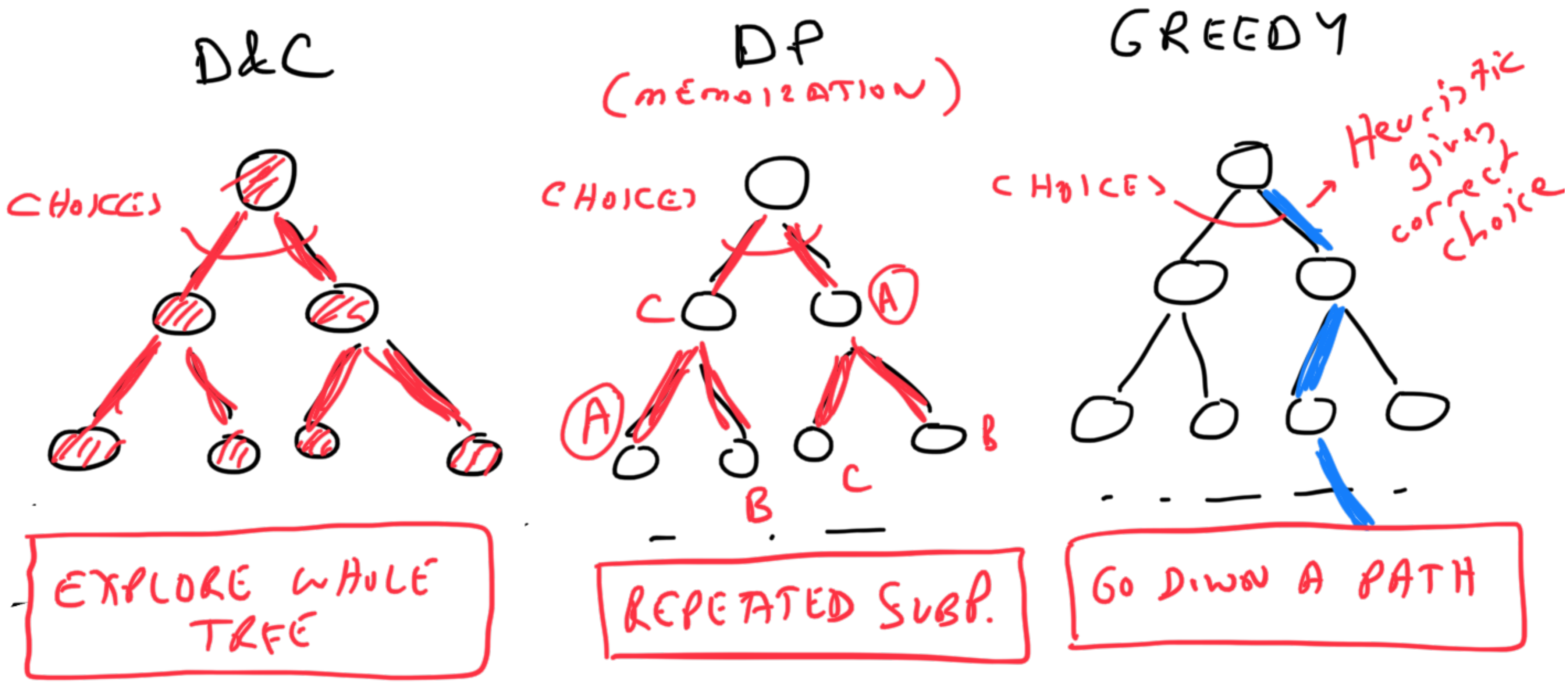
GREEDY ALGORITHMS



Common: (OPTIMAL) SUBSTRUCTURE: The solution to a problem can be expressed in terms of solutions to its subproblems.

Common: CHOICE: How do we directly

Question: Choice: how do we divide
a problem into subproblems?

ANALYSIS:

RUNTIME: D&C: Add all node merge-costs in tree.
DP: Add costs of unique nodes
Greedy: Add costs of a single path.

CORRECTNESS: D&C, DP usually obvious.

Greedy: we need to work on the proof.

OPTIMUM OR NOT?

DP that optimize: WIS etc.
that do not: Robot $(0,0) \to (m,n)$; Fibonacci
(they use Σ instead of min/max)

D&C can also optimize (no repeated subp) not optimize : merge-sort.

GREEDY EXAMPLES:

① N types of metals

$w_1 \dots w_n$: amount of i'th metal.

$c_1 \dots c_n$: price per kg i'th metal.

$k_1 \dots k_n$: sold amount in kg i'th metal.

s.t. I make $\boxed{T\ \$}$ total : $T = \sum c_i k_i$

Minimize : $\sum k_i$

=> SORT metals by price (decreasing)

for $i = 1 : n$

if $\boxed{w_i} \geq T/c_i$

$\Rightarrow$ send T/c_i kg of metal i, done

else

send all of w_i (worth $w_i c_i$)

need to send $T - w_i c_i$

continue with next metal.

PROOF: Cut-and-paste method.

Assume someone claims an optimal solution that violates your rule.

Then modify their solution according to your rule and show it improves.

Say at some point we used a cheaper metal i when a more expensive metal j was available.

$(c_j > c_i)$. We can replace $c_j k_j$ with $c_i k_i$ where $k_i = \frac{c_j k_j}{\boxed{c_i}} < k_j$ and decr weight

The solution was not optimal.

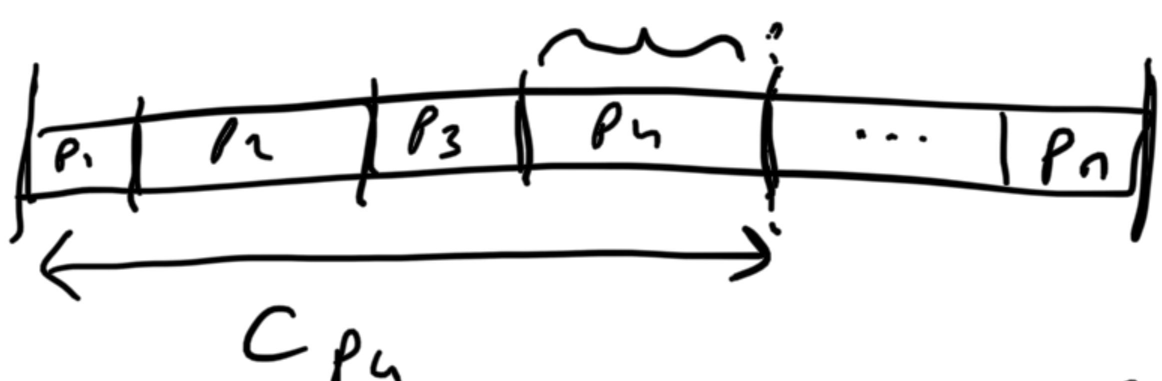
EXAMPLE 2:

n Processes.

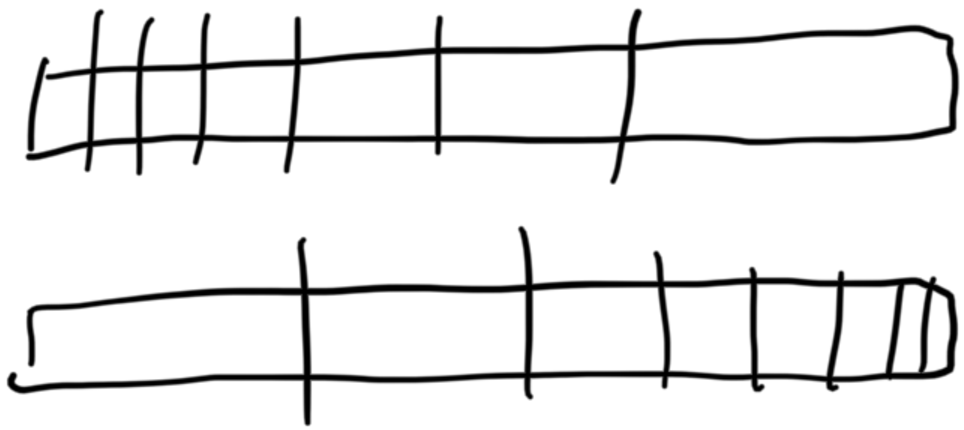
$p_1 \ p_2 \cdots p_n$: first do p_1, then p_2 ...

$t_{p_1} \ t_{p_2} \cdots t_{p_n}$: time it takes for p_i

Completion time: $C_{p_i} = \sum_{j=1}^{i} t_{p_j}$



Find an ordering p_i s.t. $\frac{\sum C_{p_i}}{n}$ is minimized



PROVE SHORTEST WORKS USING CUT & PASTE.