

cs5050

02_20_14

①

Study

"Could you adapt the linear space method?"

lean knapsack - exact Fit T or F

$$f(n) = a f(n/b) + c n^k$$

$a =$
 $b =$
 $k =$

number of
recursive calls

reduction in size

the work done per call
to split problem
merge solutions

$$\begin{aligned} \text{IF } a &> b^k \\ a &= b^k \\ a &< b^k \end{aligned}$$

$$\begin{aligned} n^{\log_b a} \\ n^k \log n \\ n^k \end{aligned}$$

1) Understand what the parameters a, b and c mean

2

Possible Question
write alg.

Solve multiply
two linear functions
degree 1 poly.

Solve

$$(p_0 + p_1 x) \cdot (q_0 + q_1 x)$$

=

- 1) Is there an efficient way to multiply two linear functions to determine the coefficients of the resulting quadratic?

③ $f(n) = 2f(n-1)$

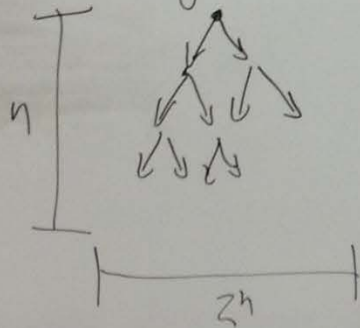
unfolding

$$= 2 \cdot 2f(n-2)$$

$$= 2 \cdot 2 \cdot 2f(n-3)$$

$$= 2^n$$

calling tree



knap sack problem
Bool knap(i, S)

base case
 $S \leq 0$ return True
 $i \leq 0 \ \& \ S > 0$ return False
 $S < 0$ return False
 check if i, S in cache
 sub problems

sol \leftarrow knap(i-1, S) ||
 knap(i-1, S-s[i]) ||
 knap(i-1, S-2*s[i])

solution construction
 store at i, S sol
 return sol

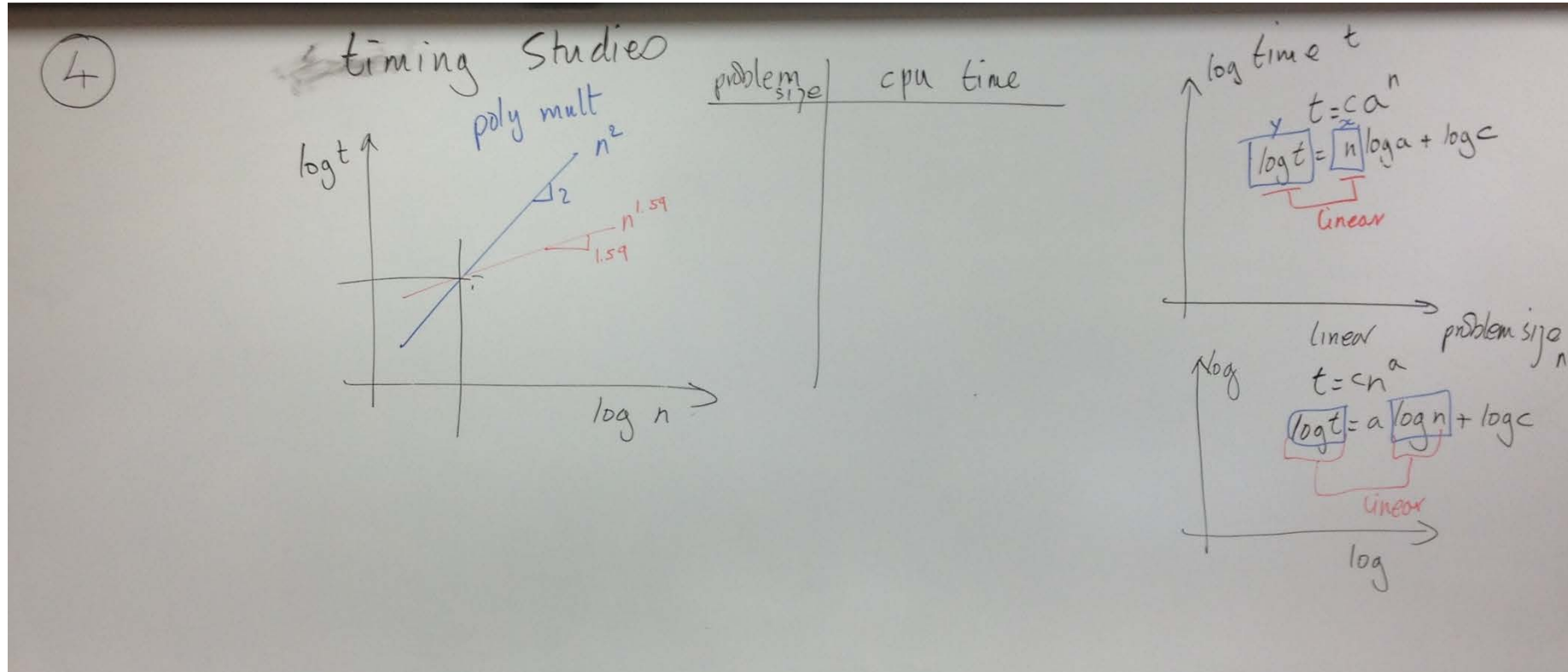
Boolean Fit problem

$s[i]$ sizes

S size

use 0, 1 or 2
 copies of the same
 object

1) Given a problem description, can you create a recursive solution? Can you then analyze it to determine the number of recursive calls it will generate? Can you identify the cache size needed? Can you create a DP algorithm from the recursive definition?



- 1) Use a linear log plot when the expected performance is ca^n
- 2) Use a log log plot when the expected performance is cn^a
- 3) Why?
- 4) What is the significance of the slope and offset of the lines on the graph?
- 5) What is the significance of the crossing point of two algorithm performances?

⑤ Evaluating a Polynomial
 $P(x) = p_0 + p_1x + p_2x^2 + \dots + p_{n-1}x^{n-1}$

$$= 2 + 5x + 10x^2 + 15x^3$$

$$P(3) = 2 + 5 \times 3 + 10 \times \underbrace{3 \times 3}_{\text{compute once}} + 15 \times \underbrace{3 \times 3 \times 3}_{\text{use twice}}$$

how many multiplies?

→ compute once use twice

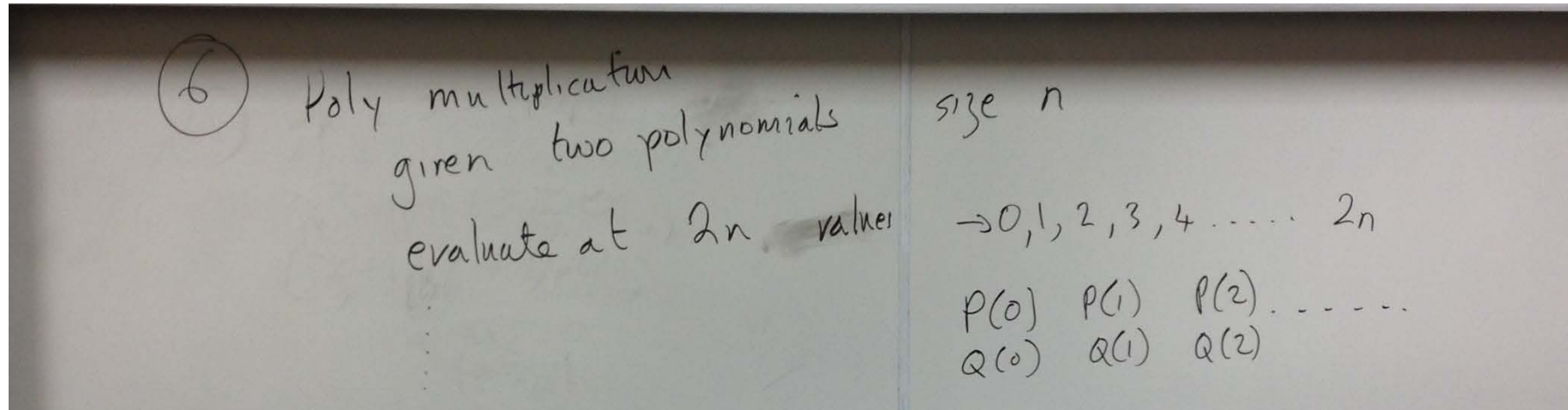
$$p(x) = p_0 + p_1x + p_2x^2 + p_3x^3 + p_4x^4$$

$$p_0 + x(p_1 + x(p_2 + x(p_3 + p_4x))) \approx n \text{ multiplies}$$

$$\begin{array}{l} 6 \sum_{i=1}^n i \approx n^2 \\ 5 \sum_{i=1}^n 2 \approx 2n \end{array}$$

1) Different ways to evaluate a polynomial of degree n

- 1) Simple way where $p_i x^i$ takes $i+1$ multiplies, needs a total of n^2 multiplies
- 2) Caching x^i to compute x^{i+1} takes multiplies total $2n$
- 3) Using the factoring technique above (add then multiply) takes only n total



- 1) New way of multiplying two polynomials P and Q with n terms each to find PQ , the polynomial that is P times Q
 - 1) First evaluate each at $2n$ values (why do we need this many?, because the result has $2n-1$ terms we need at least this many to find the unknowns. If we have $2n$ unknowns, we need at least $2n$ equations)
 - 2) Multiply each $P(j)$ by $Q(j)$ to get $PQ(j)$, where j is the value of x . We know that PQ must pass through each $y=PQ(i)$ point. We interpolate to solve for the coefficients given points on the polynomial (we have not done this step yet, so don't bother understanding it now).

Study

① "Could you adapt the linear space method?"

• Boolean knapsack: exact fit T/F

$S = 10$

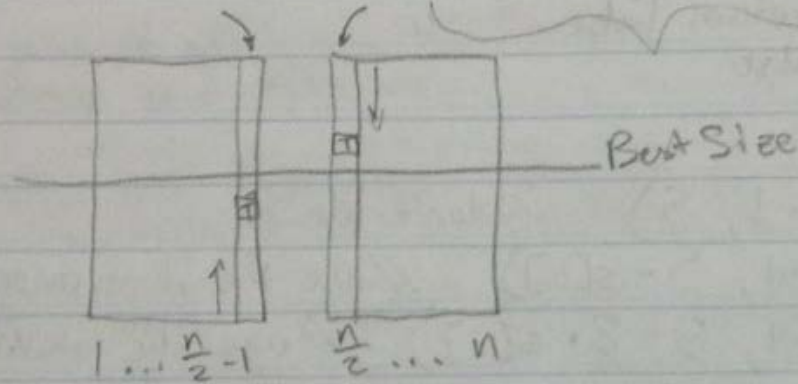
$s[1] = 5$

$s[2] = 3$

$s[3] = 7$

$s[4] = 11$

How?



1) You know how to use the linear space, D&C algorithm to find the objects in the max value knapsack problem. Can you adapt the method to find the objects in a "find a subset of the objects that exactly fit into a knapsack" Boolean problem?