ACGTCG $X_{1 \ldots n}$

AATTGC $Y_{1 \ldots m}$

i) $X_n = Y_m$ $LCS(X,Y) = LCS(X_{1..n-1}, Y_{1..m-1}) + 1$

ii) $X_n \neq Y_m$ both G & C are not last char.

$$LCS(X,Y) = \max \begin{pmatrix} LCS(X_{1 \ldots n-1}, Y) \\ LCS(X, Y_{1 \ldots m-1}) \end{pmatrix}$$

Oct 15-4:09 PM

lcs.pdf (application/pdf Object) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.columbia.edu/~cs2035/courses/csor4231.F09/lcs.pdf

7 / 8 100%

Recursion for length

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0\ , \\ c[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } x_i = y_j\ , \\ \max(c[i, j-1], c[i-1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq y_j\ . \end{cases}$$

$c[i,j]$ = length of $LCS(X_{1..i}, Y_{1..j})$

11.00 x 8.50 in

Oct 15-4:22 PM

lcs.pdf (application/pdf Object) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.columbia.edu/~cs2035/courses/csor4231.F09/lcs.pdf

Recursion for length

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0\ , \\ c[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } x_i = y_j\ , \\ \max(c[i, j-1], c[i-1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq y_j\ . \end{cases}$$

$c[i,j]$ is the length of the lcs of $X_{1..i}$ & $Y_{1..j}$

11.00 x 8.50 in

Oct 15-4:28 PM

Greedy Algorithms

- easy to design
- not always correct
- hard to identify when greedy is the right solution

Greedy alg makes its next step based only on prev. steps (current state) and "simple" calculations over the input.

Oct 15-4:28 PM

Rod Cutting is not greedy

i	1	2	3	4
profit	5	10	11	15

$(A_1 A_2)(A_3 A_4 A_5 A_6)$

Greedy: Change w/ US coins

Oct 15-4:31 PM

Room Scheduling

Alg: Pick smallest interval , recurse

Oct 15-4:37 PM

greedy.pdf (application/pdf Object) - Mozilla Firefox

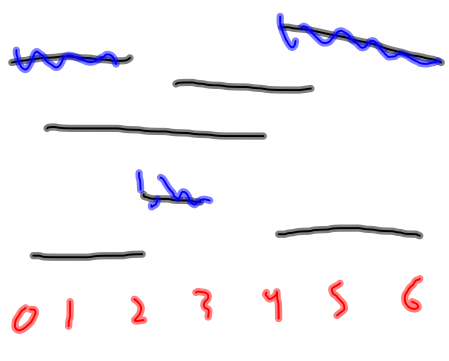
Greedy

Consider a set of requests for a room. Only one person can reserve the room at a time, and you want to allow the maximum number of requests. The requests for periods (s_i, f_i) are:

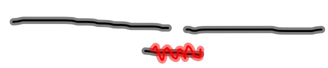
$$(1,4), (3,5), (0,6), (5,7), (3,8), (5,9), (6,10), (8,11), (8,12), (2,13), (12,14)$$

Which ones should we schedule?

Oct 15-4:39 PM



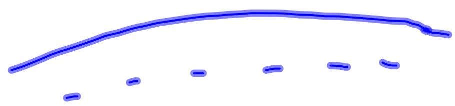
(0,7) (6,8) (7,14)



Shortest Segment First is not optimal

Oct 15-4:39 PM

X Start at end, choose last finishing segment, recurse



X Start at beginning, choose earliest start

X Pick segment w/ min # of overlaps

Oct 15-4:42 PM

Alg: Start at beginning, choose earliest finishing segment; recurse.

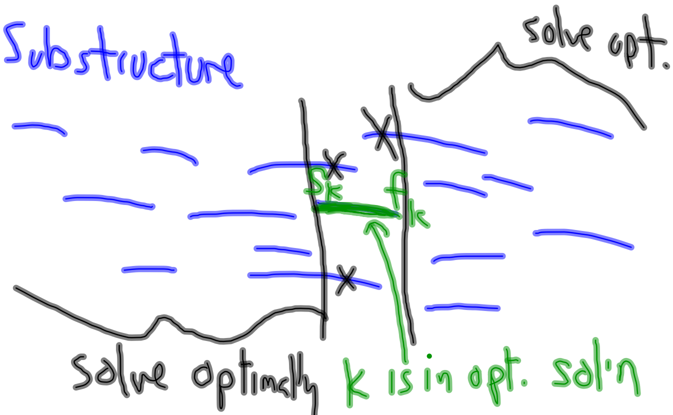
Prove a greedy alg is opt., I need

1) optimal substructure
2) greedy choice property

$\exists$ an opt sol'n consistent w/ my greedy choice in the first step.

Oct 15-4:46 PM

Opt. Substructure



- can write down opt. substr. proof
- $m[i,j]$ = # of intervals in an opt. solution from i to j

$$m[i,j] = \max_{\substack{\text{interval } k \\ \text{in} \\ (i,j)}} \begin{cases} m[i,s_k] + m[f_k,j] + 1 \\ \text{base case} \end{cases}$$

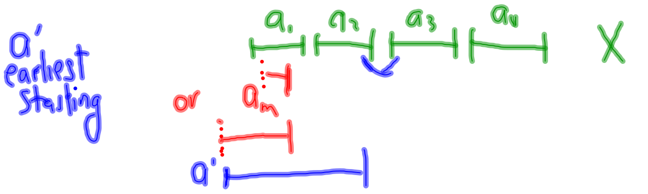
Oct 15-4:49 PM

Greedy choice property:

Let $a_m = (s_m, f_m)$ be the activity in (i,j) w/ the earliest finishing time. Then

1) $\exists$ an optimal solution using a_m
2) (i, s_m) is empty.

Pf Suppose we have an opt. solution X w/o a_m.
(we need to show how to convert X into another solution that uses a_m and is at least as good)



then $X - a_1 + a_m$ is also a valid solution (because $f_m \le s_j$ $(j \ge 2)$) and it schedules as many intervals as X ☒

Oct 15-4:58 PM

Greedy

1) Identify opt. substructure
2) Cast problem as a greedy choice & prove g.c. property
3) (Write a simple iterative alg.)

Oct 15-5:14 PM

Robbery

Rob a house, I have a knapsack. Fill the knapsack w/ the most profitable items.

item	1	2	3	knapsack capacity
weight	10	20	30	$B=50$
value	60	100	120	
val/wt.	6	5	4	

Integral knapsack - take an item or leave it

fractional knapsack - take a fraction of an item (infinitely divisible)

Oct 15-5:16 PM