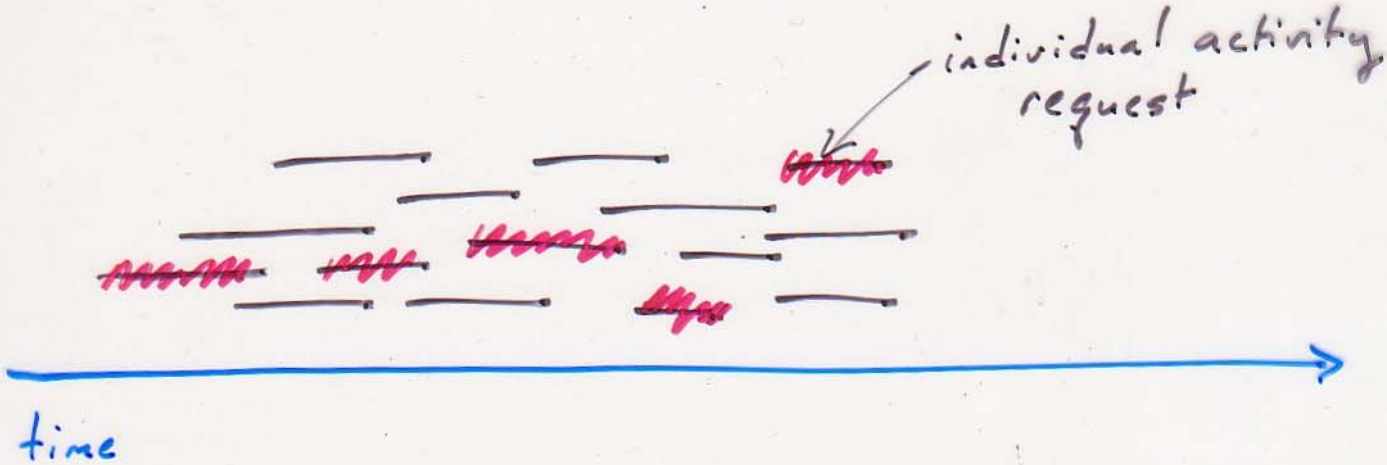
Lec 14

Greedy Algorithms

- useful for optimization problems
- locally optimal choices lead to globally optimal soln.



Problem: find the largest # of mutually compatible requests.

sort activity requests

so that

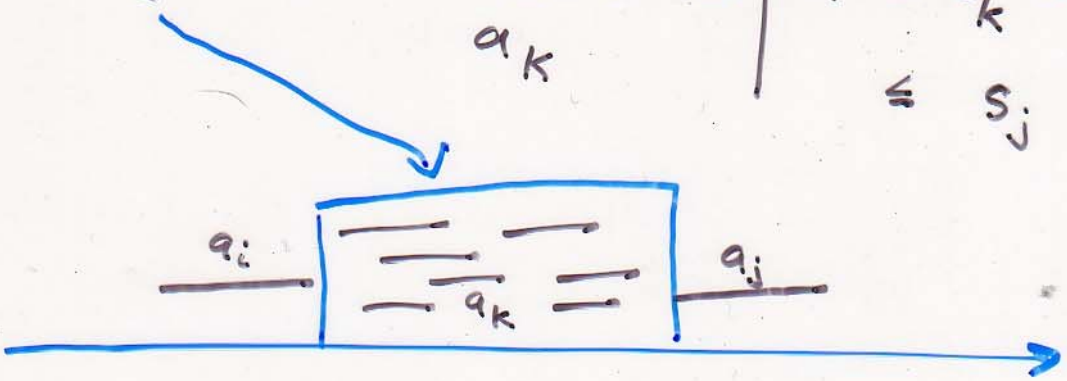
$$f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_n$$

[activity request i
denoted a_i
start time: s_i
finishing time: f_i]

define a_0: $s_0 = f_0 = 0$

a_{n+1}: $s_{n+1} = f_{n+1} = \infty$

$$S_{ij} = \{ \text{all activities } a_k \mid f_i \leq s_k < f_k \leq s_j \}$$



Observe $i > j \Rightarrow S_{ij} = \phi$

S = all activities = $S_{0,n+1}$

Let A be an optimal soln to the original problem.

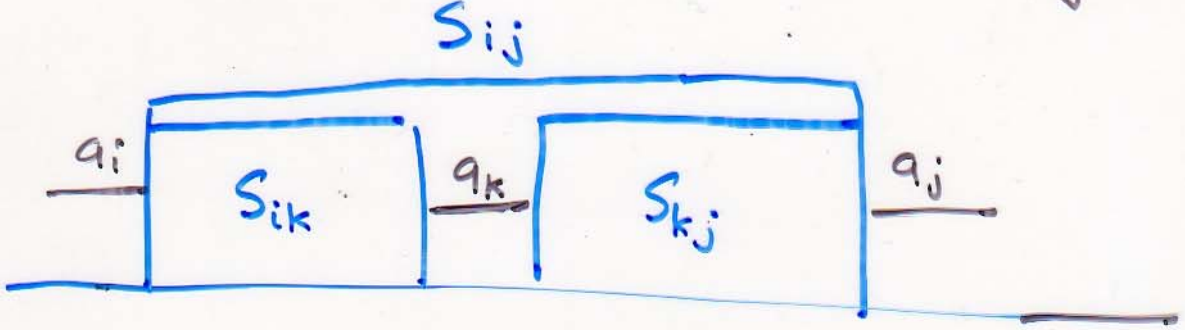
dynamic programming approach:

- define smaller problems:
  ... S_{ij}
- let A_{ij} be an optimal soln to S_{ij} (maximum # of activities to select from S_{ij})

- suppose $a_k \in A_{ij}$

then

$$A_{ij} = A_{ik} \cup \{a_k\} \cup A_{kj}$$

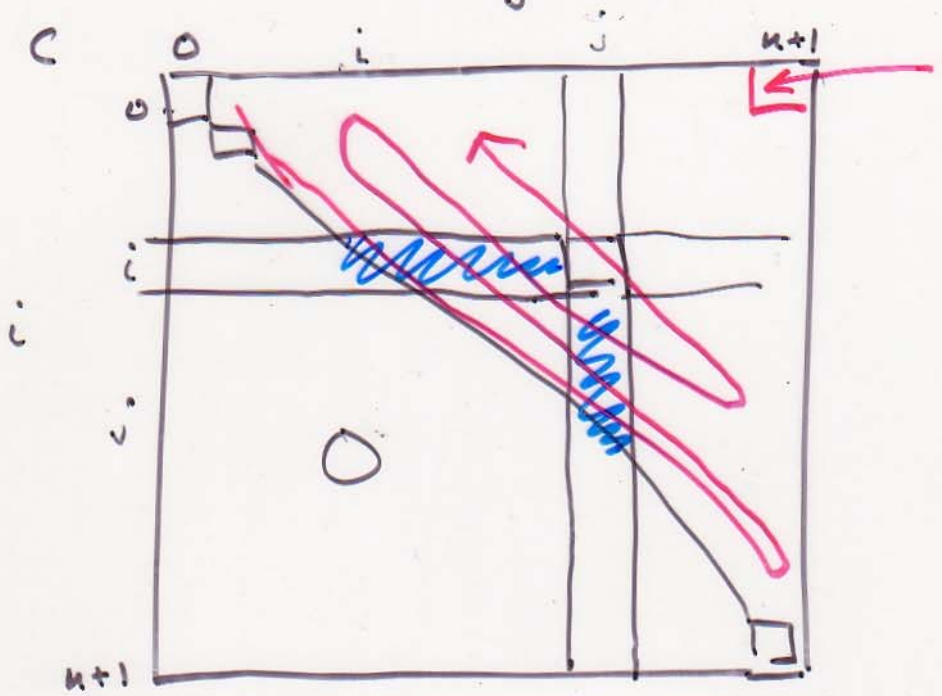


- let $c[i,j] = |A_{ij}|$

- we can express a recurrence relation

$$c[i,j] = \begin{cases} 0 & \text{if } S_{ij} = \emptyset \\ \max\limits_{i<k<j} \{ c[i,k] + c[k,j] + 1 \} & \text{if } S_{ij} \neq \emptyset \end{cases}$$

... following dynamic programming

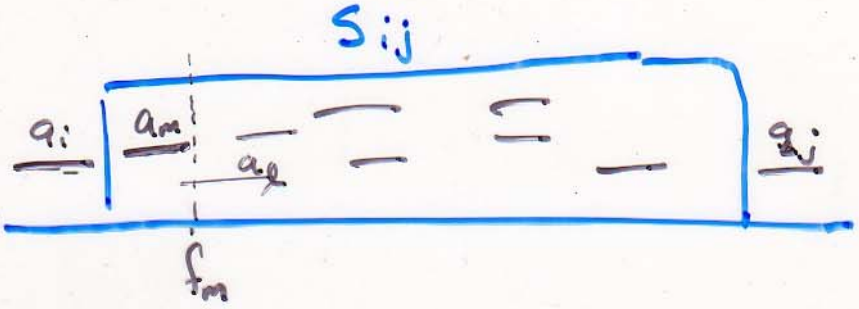


Why does a greedy strategy applies?

Let $f_m = \min \{ f_k \mid a_k \in S_{ij} \}$

Claim a_m is part of an optimal solution

$A_{ij} = \{ a_m \} \cup A_{mj}$


S i,j
q_i
q_m
q_k
q_j
f_m