CSL 356 Lect. 16 Sept 5

Basic Greedy algorithm for a Job Scheduling problem

Jobs $J_1, J_2 \ldots J_n$ each with unit processing requirement

Deadlines $d_1, d_2, \ldots d_n$

Penalty $p_1, p_2, \ldots p_n$

Obj: Maximise the penalty of the scheduled jobs

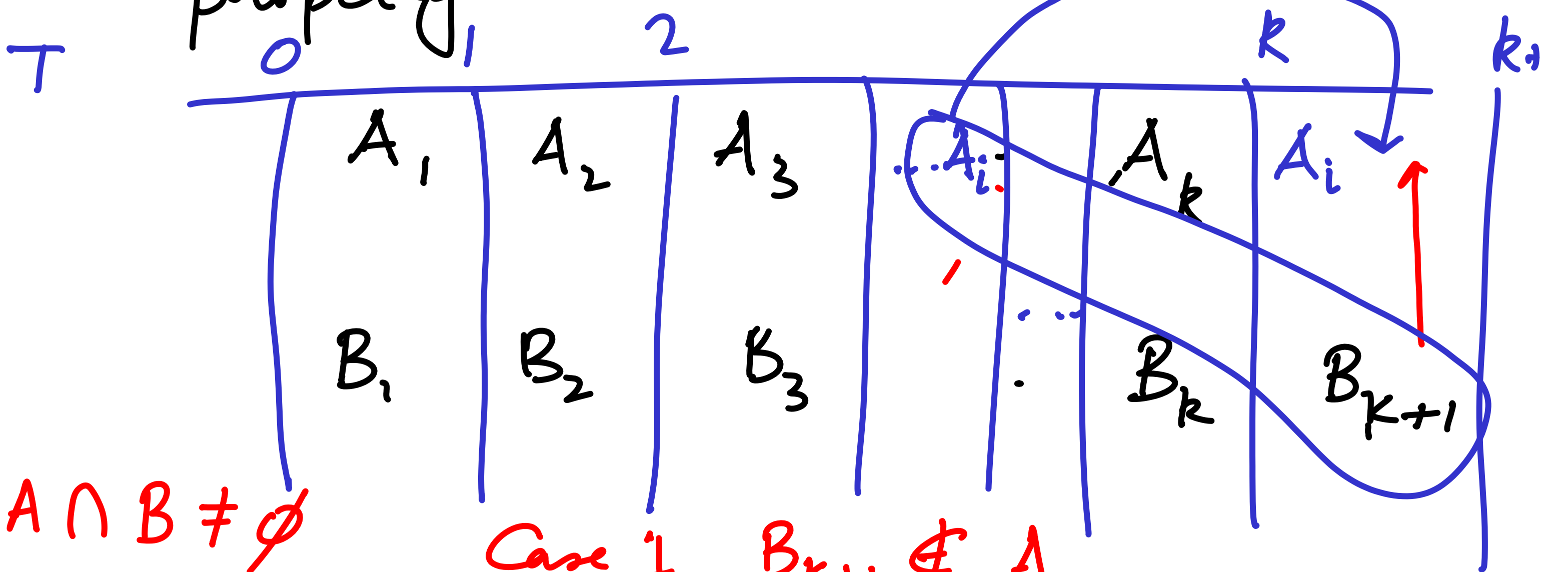
A set of jobs $J_{i_1}, J_{i_2} \ldots J_{i_k}$ is "feasible" if they can be scheduled without incurring any penalty

<u>Basic Greedy</u>

Starting from the largest penalty job keep adding the next highest penalty-incurring job (if feasible).

Does basic greedy yield optimal soln?

We will try to prove the exchange property

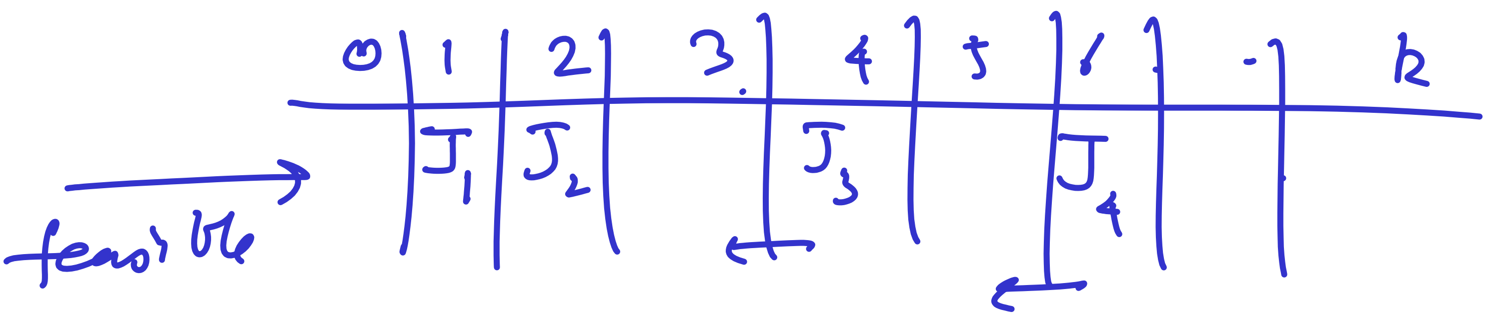


$A \cap B \neq \emptyset$

Case I $B_{k+1} \notin A$

The jobs are given in order of some feasible schedule.

(It is an algorithmic problem to determine a feasible schedule)



gaps can be compressed to left

Case II $B_{k+1} = A_i$ for some $i < k+1$

Move A_i to the interval $k, k+1$

Look at the set of jobs ignoring the last col.

Can we add the next most profitable element and maintain feasibility?

(S, M) ← family of "feasible subsets"

and M can be very large

$M \subset 2^S$, so maintaining M explicitly will be extremely inefficient.

Instead we characterise the subsets of M using some <u>property</u>

→ Maximal Spanning trees: no cycles

(Does the matroid theorem extend to minimisation function, specifically Minimal Spanning <u>Trees</u>)

At any stage of the MSF problem we have a set of trees. We add the next edge if it doesn't induce a cycle

<u>Obs</u> We can add the next edge (x, y) iff x and y belong to different connected components

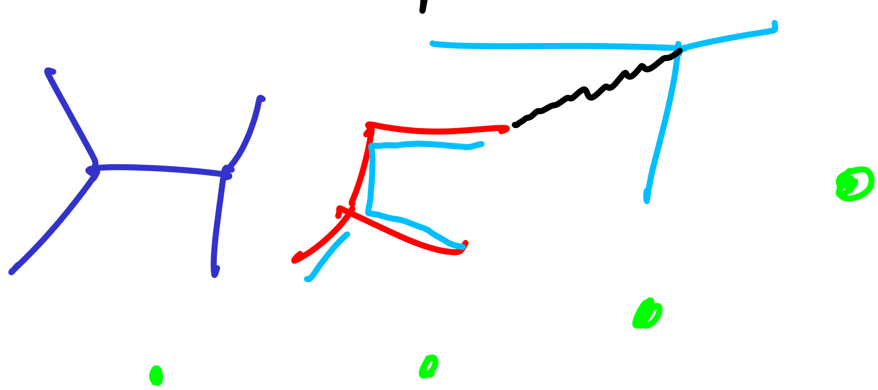
How quickly can we do this test?

What is the right data structure

$C(\ )$: defines the component

$C(x) \stackrel{?}{=} C(y)$?

If $C(x) \neq C(y)$ then we must add the edge and combine the components



Label the vertices with the component nos (initially, 1, 2, ... n)

When you join, change the labels of one component

Test $\quad C(x) = C(y)$

Find $\quad O(1)$ time

Join : $O(\min(|C_x|, |C_y|))$

Union

size of smaller component

What is the overall cost for

$\boxed{2m}$ tests $\quad \boxed{n-1}$ Joins ?

$m = |E| \qquad |V| = n$

m Finds and n unions

$O(m) \qquad\qquad O(n^2)$?

How often does a specific vertex change its label.