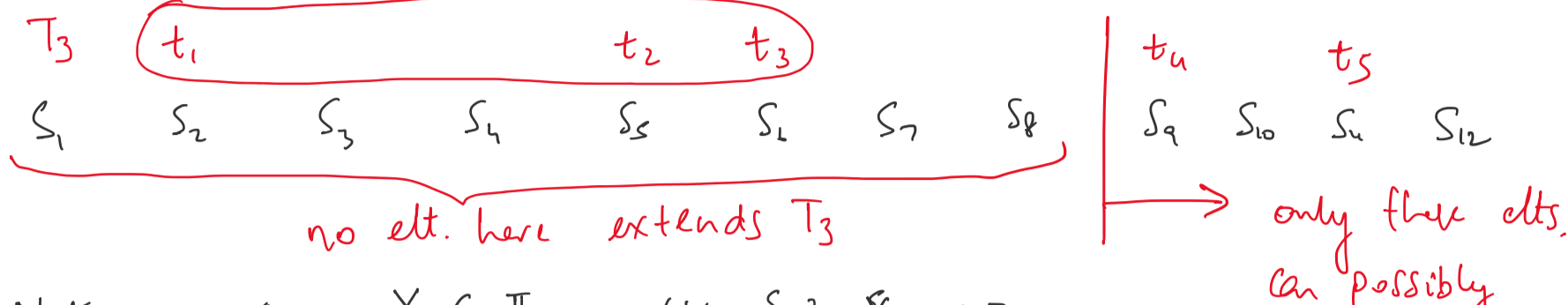


Problem: Given matroid $M = (S, \mathbb{I})$ with wt. $w(s_i)$ on each element $s_i \in S$, find an independent set of max. wt. All wts. are nonnegative.

Algo: Sort elts. so that $w(s_1) \geq w(s_2) \geq \dots \geq w(s_n)$
 $T \leftarrow \emptyset$
 for $i = 1 \dots n$
 if $T \cup \{s_i\} \in \mathbb{I}$, $T \leftarrow T \cup \{s_i\}$

Claim: T obtained at the end of the algo is an IS of max. wt.

Proof: Let $T = \{t_1, t_2, t_3, \dots, t_k\}$ in decreasing order of wt.



Note: ① say $X \in \mathbb{I}$, $a \notin X$, $\{a\} \cup X \in \mathbb{I}$.

Then for any $Y \subseteq X$, $\{a\} \cup Y \in \mathbb{I}$

② no elt. before t_i extends T_{i-1} (in example above $i=4$)

Let $T_i = \{t_1, t_2, t_3, \dots, t_i\}$, i.e., T_i is the first i elts of T .
 Actually will show that for all $i \leq k$, T_i is the highest wt. IS of size i .

Base case: clearly true for $i=0$

Suppose T_{i-1} is the max wt. IS of size $i-1$

For a contradiction, assume $w(\hat{T}_i) > w(T_i)$

By exchange property, $\exists a \in \hat{T}_i \setminus T_{i-1}$ s.t. $T_{i-1} \cup \{a\} \in \mathbb{I}$.

But then, a must lie after t_i , or $w(a) \leq w(t_i)$ by Note 2

also, $w(\hat{T}_i \setminus a) \leq w(T_{i-1})$ by induction hypothesis

Thus $w(\hat{T}_i) \leq w(T_{i-1} \cup t_i) = w(T_i)$

Do yourself: Now suppose some elements have negative wt. How do you modify algorithm?

OR: if you want max wt. IS of size k , in presence of -ve wts.?

OR: if you want min wt. IS?

Problem: Job Selection

Given: n jobs J

deadlines $d_1, d_2, d_3, \dots, d_n$

penalties $w_1, w_2, w_3, \dots, w_n$

Each job takes 1 unit of time to finish processing. We have a single machine

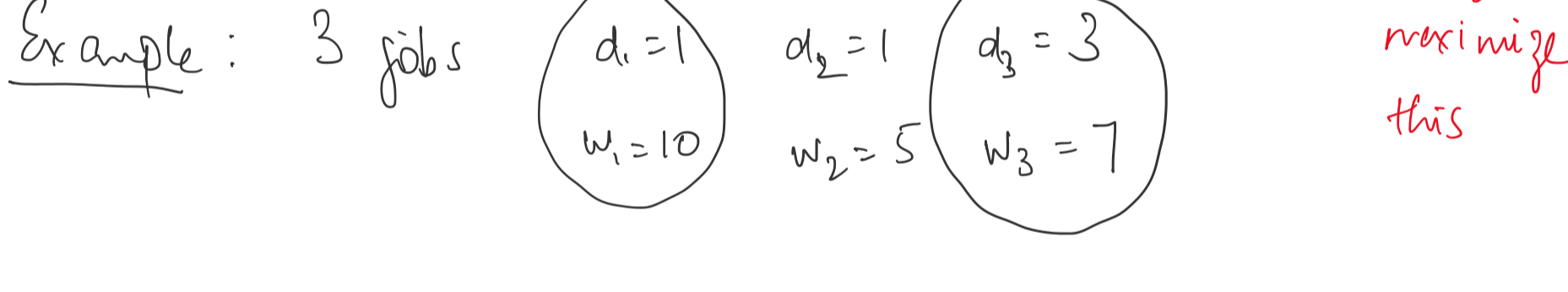
Defn: Given $S \subseteq J$, a schedule is an ordering of the jobs in S

Given $S \subseteq J$ a feasible schedule is an ordering where each job completes ^{on or} before its deadline

Given $S \subseteq J$, S is a feasible set if S has a feasible schedule

Problem: Find a feasible set $S \subseteq J$ that

minimizes $\sum_{j \notin S} w_j = \sum_{j \in J} w_j - \sum_{j \in S} w_j$



Let $D := \max d_j$

For any $t \leq D$, given $S \subseteq J$, $N_t(S) = \#$ of jobs in S that have deadlines $\leq t = \{j \in S: d_j \leq t\}$

Lemma: Given $S \subseteq J$; the following statements are equivalent:

- ① S has a feasible schedule
- ② For any $t \leq D$, $N_t(S) \leq t$
- ③ The schedule that orders jobs in S by increasing deadlines is feasible

Proof: ③ \Rightarrow ① : trivial
 ① \Rightarrow ② : by contradiction
 ② \Rightarrow ③ : easy. (do yourself)

Lemma: Let $\mathbb{I} = \{S \subseteq J: S \text{ is feasible}\}$. Then $M = (J, \mathbb{I})$ is a matroid.

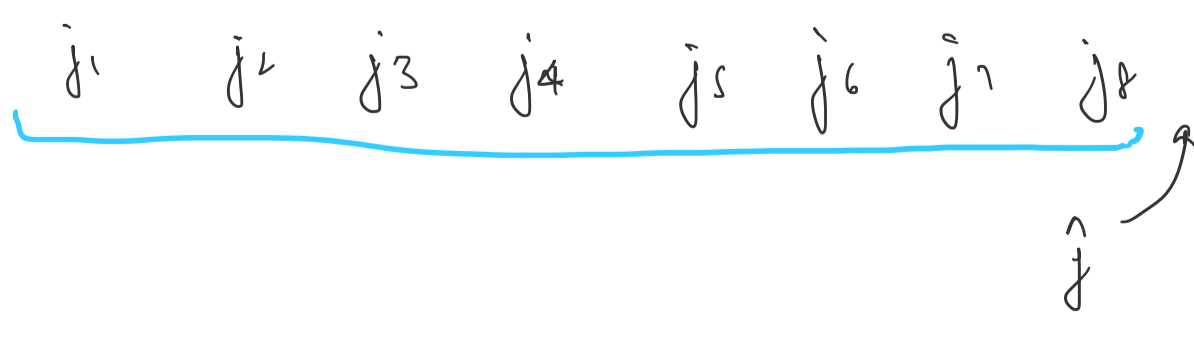
Proof: ① Downward-closed is trivial, if $S \subseteq J$ is feasible then so is any subset

② Let $S, T \subseteq J$ s.t. both are feasible & $|S| > |T|$.

Let $\hat{j} \in S$ be the job w/ the last deadline

Consider $T \cup \hat{j}$.

Say job in T in order of increasing deadline are



Claim: The schedule that puts jobs in T in order of increasing deadline & \hat{j} at the end is feasible

Proof: $d_{\hat{j}} \geq |S| > |T|$, hence a schedule since S is feasible \hat{j} after all jobs in T .

