29 January -	- Greedy Sta	ys Ahead	· · · · · · ·	· · · · · · · · · ·
$= \frac{1}{2} \left[\frac{1}{2} \left[\frac{1}{2} \right] + \frac{1}{2} \left[\frac{1}{2} \left[\frac{1}{2} \left[\frac{1}{2} \right] + \frac{1}{2} \left[\frac{1}{2} \left$	· · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
$\frac{1}{2} \frac{1}{2} \frac{1}$		· · · · · · · · ·	· · · · · · ·	· · · · · · · · · ·
(1). Interval Schedu	Nice Problem	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
(2) Announcements	· · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
3 Greedy Stays	Ahead	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·	
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · ·	· · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · ·	· · · · · · ·	· · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · ·	· · · · · · ·	· · · · · · · · ·

· · · · · · · · · · · · · · · · · · ·		assi ESì	ngle		Cel	i or htral		· · · · · · · · · · · · · · ·		2 S S'				· · · · · · · · · · · · · · · · ·	 . .<	 . .<	· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· ·		ves	tio		• •	Hon	· · ·			e :	Śċ	he.	Jul	۰۰۰ بر ۲	H	12		65		• •
· ·		• •	· · ·	· ·	• •	· · ·	• •	• •	· · · ·	• •		• •	· · ·	· ·	• •		· · · ·	• •	• •	• •
• •	· · ·	· ·	· · · ·	· ·	· ·	· · ·	• •	· ·	· · · ·	• •	· ·	· ·	· · · ·	· ·	· ·	· ·	· · · ·	· ·	• •	• •
• •	· · ·	· ·	· · ·	· ·	· ·	· · ·	· ·	· ·	· · · ·		· · ·	· ·	· · ·	· ·	· ·	• •	· · · ·	· ·	• •	• •
••••		· ·	· · ·	· ·	· ·	· · ·	· ·	· ·	· · · ·		· · ·	· ·	· · ·	· ·	· ·	· ·	· · · ·	· ·	• •	••••
• •					· ·							· ·								
• •				• •		· · ·	• •	• •						• •	• •	• •			• •	
• •	· · ·	· ·	· · ·	· ·	• •	· · ·	• •	· ·	· · ·	••••	· ·	• •	· · ·	• •	· ·	• •	· · ·	· ·	• •	• •
· ·	· · ·	· ·	· · ·	· ·	· ·	· · ·	• •	• •	· · · ·		• •	· ·	· · ·	· ·	· ·	• •	· · · ·	· ·	· ·	· ·

Classic Motivation: * single central processor * many job requests Question: How do we schedule the jobs? Details - Each job has a proposed start time & finish time - Processor can handle at most 1 job at a time - Assumption. j>bs have equel priority

Classic M * single * many	central processor job requests	
(Vuestion Start 1 start 1	How do we schedule finish 1	the jobs. finish 2
J°BS 	stut3 finish 3 start n	finish n

Inter	rval Scheduling Problem
Given	List of n jobs, specified by [start, finish] time
. 	$\left\langle \left[S_{1}, f_{1} \right], \left[S_{2}, f_{2} \right], \ldots, \left[S_{n}, f_{n} \right] \right\rangle$
Goal	Return a set of non-conflicting jobs
· · · · · · ·	of maximum cardinality
	· · · · · · · · · · · · · · · · · · ·
· · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · ·	<pre></pre>
. 	1 1
	1 1
. 	1 1

Interval Scheduling Problem
Given List of n jobs, specified by [start, finish] time
$\left\langle \left[S_{1}, f_{1} \right], \left[S_{2}, f_{2} \right] \right\rangle$
Goal Return a set of non-conflicting jobs
of Maximum cardinality
Equal priority
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

Interval Scheduling Problem Given List of n jobs, specified by [start, finish] time $\left\langle \left[S_{1},f_{1} \right],\left[S_{2},f_{2} \right],\ldots,\left[S_{n},f_{n} \right] \right\rangle$ Goal Return a set of non-conflicting jobs of maximum cardinality Defn. A set of intervals S is <u>non-conflicting</u> if for all $i \neq j \in S$ $S_i \leq S_j \implies f_i < S_j$ fin fin fin hon-conflicting

Interval Scheduling Problem Given List of M jobs, specified by [start, finish] time $\left\langle \left[S_{1},f_{1}\right] ,\left[S_{2},f_{2}\right] ,\ldots,\left[S_{n},f_{n}\right] \right\rangle$ Goal Return a set of non-conflicting jobs Defn. A set of intervals S is <u>non-conflicting</u> if for all $i \neq j \in S$ $s_i \leq s_j \implies f_i < s_j$ fi finner conflicting Sk (free conflicting)

•	· ·	А. - С. - С.	N. I	n Ni () Di C	i ĻV			> (M	e	N	i-f	-"\	· · ·	•	•	•	•	•	•	•	•	•	•	•	• •	• •		•	•	•	• •	•	•	•	•			· ·			•
•	• • • •	, , , ,	•	ŀ	- \/			b D	•	•			رو ر	•	•	•	•		· ~ .	Se	5.(A	· · ·				J.		T·V			· /· · ·		•	•	•	•	· · ·		•	• • •
•	•••	X			`\ \	Ú	•	-1	• •	•	K		2 (2	عع	بع	J	•	•	Å		÷.	د	 	•		e R (<u>-</u> -		اهر	•		• •	•	•	•	•	•	•	· ·		•	•
•	• •	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	0	0	•	•	•	•	•	•	•	•	• •	• •	•	о Ф	•	•	• •	•	•	•	•	•	• •	• •	•	•	•
•	•••	•	•	•	•	0 0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	• •	•	•	•	•	• •	•	•	•	•	•	•	• •	•	•	•
•	• •	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	• •	•	•	•	•	••••	•	•	•	•	•	•	• •	•	•	•
•	• •	•	0	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	•	0 0	•	• •	• •	•	•	e e	•	• •	•	0	•	•	0	•	• •	•	•	•
•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •		•	•	•	•		• •	•	•	•
•	•••	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	•	•	•	•			•		•		• •	• •		•	•		• •	•	•	•	•			• •			÷
•	· ·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•			•	•	•	•	• •	· ·	•	•	•	•	• •	•	•	•	•		•	• •	•	•	•
•	· ·	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	· ·	•	•	•	•	· ·		•	•	•		•	• •	•	•	•
	· ·		•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	· ·	•		•	•	· ·		•	•			•	· ·	•	•	•
•	• •	٠	٠		٠			•	٠	•		٠	•	•	٠	•		•	٠	٠	٠	٠		•	•		•		•	٠		٠				٠	•	0	•	• •	•	٠	٠

Inte	rval Schedulin	g Problem	· · · · · · · · · · · · · · · · · · ·
Given	List of Mijol	os, specified	by [start, finish] time
. 	$\left\{\begin{array}{cccccccccccccccccccccccccccccccccccc$	$[S_2, f_2], [S_n]$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
Goal	Return a set	of Non-confli	cting jobs
· · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	maximum cardinality
· · · · · · ·	· · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
· · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
		· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·

Interval Scheduling Problem Given List of n jobs, specified by [start, finish] time $\left\langle \left[S_{1}, f_{1} \right], \left[S_{2}, f_{2} \right] \right\rangle$ Candidate "Priority" Functions, · (+ = 0 ·

•	•	•	•		•			• •		٠		•	•	• •	٠			•		۰	•	•		•	•		• •	•		•	•		•	• •	•	٠	• •	٠
٠	•	٠	•		•					٠	٠	٠	•	• •			•		• •	٠				•	٠	•	• •	٠	•	*	• •	٠	٠	• •		•	• •	•
								• •			0		•	• •					• •		٠	•		•	0		• •			•						٠	• •	٠
	٠	•	•		٠	•	٠	• •	٠	٠	۰		•	• •				٠	• •	٠	٠	•		•	٠	•	• •		٠	•			٠	• •	•	٠		•
•		•	•			•	•		•				•	• •				٠				•		•		•	• •		٠	•					•	٠		
								• •					•	• •					• •		٠	•		•			• •			•			٠					٠
	•	•	•		•		•		•	•	٠		•	• •				٠	• •	•	•			•	٠		• •			•			٠		•			
		•				•							•					•				•		•		•			٠	•					•			
		•	•										•	• •				•			٠	•		•		•				•			٠		•			
•	•	•	•		٠	•	•		٠	٠			•	• •				*	• •	٠		•		•	*	•	• •	•	٠	•			٠		*	٠		
		•											•					•				•		•		•				•					•			
		•	•										•	• •				•						•		•				•			٠		•			
•		•	•		•		•		•		•	•	•	• •				•		•				•						•			•	• •	•		• •	
		•	•										•	• •				•				•		•		•				•					•		• •	
		•	•										•	• •				•						•						•			٠		•			
•	•	•	•	• •	٠		٠		٠	•	•	•	•	• •		٠		٠		٠				•			• •	•		•			•	• •	•		• •	
٠	•	•	•		٠					•	٠	٠	•	• •	•	۰	•	•		٠			• •	•			• •	٠		•		•		• •	•		• •	
٠	٠	•	•	•	٠				٠	٠		٠	•	• •			•	0	• •		*			•	*	•	• •	٠	٠	*	• •	٠	٠	• •		٠	• •	*
٠	٠	٠	• •		٠	•	٠	• •	٠	٠	٠	٠	•	• •	٠	٠	٠			٠	٠	•	• •	•	٠	•	• •	٠	•	•		٠	٠	• •		•	• •	٠
٠	•		•		٠	•		• •		٠		٠	•	• •	٠		٠	0	• •	٠	٠	•		•	٠	٠	• •	٠	٠	•		٠	٠	• •		٠	• •	٠
٠	٠	•	•	• •	٠		٠		٠	٠		٠	•	• •				0	• •		*	•		•	+	•	• •	٠	٠	*	• •		٠	• •		٠	• •	
٠	•		• •		٠	٠	٠	• •	٠	٠		٠	•	• •	٠	٠	٠			٠	٠	•	• •	•	٠	٠	• •	٠	•	•		٠	٠	• •		•	• •	٠
٠	•	0	•		٠	•		• •	•	٠		٠	•	• •	•		٠	0	• •		٠	•		•	٠	•	• •	۰	٠	•		٠	٠	• •	0	٠	• •	٠
٠	٠	٠	•	• •	٠	•			٠	٠		•	•	• •				0	• •		*	•		•	٠	•	• •	٠	٠	*	• •		٠			٠	• •	*
		•	•			•							•	• •			•	•			•	•		•		•		•		•					•			•
	•	•	• •		•		•		•	•	•	•	•	• •		٠	•	•	• •	•			• •	•			• •	•		•			•	• •	•		• •	
•	•	•	•	• •	٠	•	٠	• •	٠	٠	•	•	•	• •	٠	٠	•	٠		٠	•			•	•		• •	•		•		•	•	• •	•		• •	
٠	•	٠	•		٠	•	٠	• •	•	٠	٠	٠	•	• •	٠	٠	•	٠		٠	٠			•	٠		• •	٠		•		٠	٠	• •	•		• •	•
•	•	•	•		•		•	• •	•				•	• •				•	• •	•				•			• •			•			٠		•		• •	•
•	•	•	•	• •	٠		٠		٠	•	•	•	•	• •	•	٠		٠	• •	٠				•	•		• •			•		•	•	• •	•		• •	
	•	•	• •		•		•	• •	•	•	•		•	• •			•	•			•	•		•		•	• •	•		•			•	• •	•		• •	•
•	•	•	•		•		•	• •	•	•	•	•	•	• •		•		•	• •	•				•			• •		٠	•			٠		•		• •	•
•		•	•			•	•		•			•	•	• •			•	•				•		•		•	• •	•	٠	•		•			•	•	• •	٠
•	٠	•	•		٠	•		• •	٠	٠			•	• •				•	• •		٠	•		•		•	• •	•	٠	•			٠	• •		٠	• •	٠
٠	*	•	•	• •	٠	•	٠	• •	٠	٠	٠	•	•	• •		٠		٠	• •	٠	٠	•		•	٠	•	• •	•	٠	*	• •		٠		•	٠	• •	•
٠	•		•		٠	٠		• •		٠		٠	•	• •	٠		٠			٠	٠	٠	• •	•		٠	• •		٠	•		٠	٠	• •		٠	• •	٠
						•							•	• •			•			•		٠		•						•					•			

Earliest Finish Time D Sort jobs by finish time 2) Schedule = 25 Iterate through jobs in sorted order j=1....n - if job j does not conflict w/ Schedule -> Schedule <- Schedule vZjJ Return Schedule.

Earliest Finish Time D Sort jobs by finish time (2) Schedule = 255 Iterate through jobs in sorted order j=1....n if job j does not conflict w/ Schedule -> Schedule <- Schedule vZjG Return Schedule.

Earliest Finish Time D Sort jobs by finish time (2) Schedule = Zign Iterate through jobs in sorted order j=1....n - if job j does not conflict w/ Schedule -> Schedule <- Schedule vZjG Return Schedule. Claim. EFT can be implemented with RT O(nlogh),

Theorem EFT returns a maximum cardinality set of non-conflictive jobs. * Non-conflicting -> By design * Maximum Cardinality -> "Greedy stays ahead

Greedy St	tays ahead	· · · · · · · · · · · ·	. .	
- Imagi	ine some	optimal so	chedule S*	
- $Compa$	ave output		$- \frac{1}{2} + $	· · · · · ·
· · · · · · · · · · · ·	· · · · · · · · · · · ·			
· · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·
· · · · · · · · · · · ·	· · · · · · · · · · · ·			
· · · · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·
· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · ·	
· · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · ·
		· · · · · · · · · · · ·		· · · · · ·
· · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · ·
· · · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·
	· · · · · · · · · · · ·			

Greedy Stays ahead - Imagine some optimal schedule S* - Compare output of EFT to St Arque that: (a) if EFT "stays ahead" of St, then EFT is also optimal (b) EFT "stays ahead"

Greedy Stays ahead - Imagine some optimal schedule S* - Compare output of EFT to St Arque that: (a) if EFT "stays ahead" of St, then EFT is also optimal (b) EFT "stays ahead" WARNING: Need to define "stays ahead" per problem

Claim. EFT "stays ahead of any S* in the finish time of the jth jb $\sum_{i=1}^{n} \frac{1}{1} \sum_{i=1}^{n} \frac{1}{1} \sum_{i$ f_{i} Convenient Notation. Assume jobs ave sorted by finishing time. $f_1^* < f_2^* < - < f_n^*$

Claim. EFT "stays ahead of any S* in the finish time of the jth pb S_{1} · · · · · · · · · · · · · · · · For each i, the ith scheduled $f_i \leq f_i$ j=bs satisfy

Greedy Stays Ahead Lemma. Suppose EFT returns a schedule of k jobs. For any optimal schedule St For all $i \in [1, ..., k]$ $f_i \leq f_i^{\pi}$ "EFT stays ahead of S*

Greedy Stays Ahead Lemma Suppose EFT returns a schedule of k jobs. For any optimal schedule St, $f_{2} \leq f_{2}$ For all i e ZI,..., k J "EFT stays ahead of S* Earliest Finish Time Lemma St is optimal => EFT is also optimal. AND $\forall i$ i f i f i i f i i f i f i f i"IF EFT stays ahead of St, then EFT is optimal"

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty$ For all i e ZI,..., k J Pf. By induction on intervals added by EFT

Greedy Stays Ahead Lemma. EFT vs. St
For all $i \in \mathbb{Z}_1, \dots, \mathbb{K}_j$ $f_i \leq f_i^*$
Pf. By induction on intervals added by EFT
Base case, i=1.
* By EFT rule, first job selected because.
$f_1 \leq f_j$ for all jobs $j \in [n]$,
$\implies f_{1} \leq f_{1}^{*}$ for any choice of S^{*}
· · · · · · · · · · · · · · · · · · ·

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-\infty}^{\infty} dx = \int_{-\infty}^{\infty} dx = \int_{-\infty}^{\infty} \int_{$ For all i e ZI,..., kg Inductive Step $\left(\forall i \in \overline{f_{1, \dots, k-1}} \right) \implies f_{k} \leq f_{k}^{*}$ Inductive Hypothesis

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-\infty}^{\infty} dx = \int_{-\infty}^{\infty} d$ For all i e ZI,..., k J Inductive Step $\left(\begin{array}{c} \forall i \in \mathbb{Z} | , - , k - 1 \end{array} \right) \implies f_i \leq f_i^{\dagger} \\ f_i \leq f_i^{\dagger} \end{array} \right)$ Inductive Hypothesis $f_{\mathcal{K}} \stackrel{\alpha}{\to} \leq \stackrel{\alpha}{\to} f_{\mathcal{K}} \stackrel{\alpha}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to} \stackrel{\sigma}{\to}$ * Consider the Kth job of St Sk. Jk.

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-\infty}^{\infty} dx = \int_{-\infty}^{\infty} d$ For all i e ZI,..., kg Inductive Step: $\begin{cases}
\forall i \in \mathbb{Z}, \dots, k-1 \\
f_i \leq f_i \\
\end{bmatrix} \implies$ Inductive Hypothesis $\int_{\mathcal{K}} \frac{1}{2} \sum_{k=1}^{n} \frac{1}{2} \int_{\mathcal{K}} \frac{1}{2} \sum_{k=1}^{n} \frac{1}{2}$ * Consider the Kth job of St Sk. Jk. * By non-overlapping: fx-1 < 54

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1$ For all i e ZI,..., k J Inductive Step $(\forall i \in \mathbb{Z}_1, \dots, k-1\mathbb{Z})$ $f_i \leq f_i^*$ Inductive Hypothesis $\int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} \leq \int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} + \int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} = \int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} + \int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} = \int_{\mathbf{k}}^{\mathbf{k}} d\mathbf{x} + \int_{\mathbf$ Ska a a a free a * Consider the Kth job of St * By non-overlapping: fx-1 < sk * By IH, $f_{k-1} \leq f_{k-1}^*$ => [Su, fn] Non-conflicting w/ [Su-1, fu-1]

Greedy Stays Ahead Lemma. EFT vs. St $\int_{-\infty}^{\infty} dx = \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} dx = \frac{1}{\sqrt{2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2}} \int_{-\infty}^{\sqrt$ For all i e ZI,..., k J Inductive Step $\left(\begin{array}{c} \forall i \in \overline{2}1, \dots, k-1\overline{2} \\ f_i \leq f_i^{\dagger} \end{array} \right) \longrightarrow$ Inductive Hypothesis $\int_{\mathcal{W}} f_{\mathcal{W}} dx \leq \int_{\mathcal{W}} f_{\mathcal{W}} dx = \int_{\mathcal{W}} f_{\mathcal{W}} dx$ S_{k} S_{k * Consider the Kth job of St * By non-overlapping: fx-1 < sk * By IH, $f_{k-1} \leq f_{k-1}^*$ $f_{k-1} \sim f_{k-1} \sim f_{k$ (Su, fu] Non-conflicting w/ (Su-1, fu-1) But [skifn] is non-conflicting job of earliest finish time. fu = fu

Earliest Finish Time Lemma St is optimal => EFT is also optimal. AND $\forall i \quad f_1 \leq f_1$

Earliest Finish Time Lemma St is optimal => EFT is also optimal. AND $\forall i \in \mathcal{F}_{1}$, $\leq i \leq i \in \mathcal{F}_{1}$ Pf. Consider any non-overlappilg set St We show. EFT "stays ahead" of $S^* \implies |EFT| \ge |S^*|$ * Consider any prefix of indices 1,..., i < |EFT|

Earliest Finish Time Lemma St is optimal => EFT is also optimal. AND $\forall i f_i \leq f_i$ Pf. Consider any non-overlappilg set St We show. EFT "stays ahead" of $S^* \implies |EFT| \ge |S^*|$ X Consider any prefix of indices 1,..., 2 < |EFT| $|f| |S^{*}| \ge it|$, then $\mathcal{K} = \int_{\mathcal{I}} \int_{\mathcal{I}} \mathcal{L} = \int_{\mathcal{I}} \int_{\mathcal{I}} \mathcal{L} = \int_{\mathcal{I}} \mathcal{L} = \mathcal{L}$ [Siti , fiti] is feasible for EFT. \Rightarrow $|EFT| > |S^*|$

Recap
* Earliest Finish Time algorithm
returns maximum non-overlapping Schedule
X Runs in O(n log n) fine
X Proof off Correctness:
EFT stays ahead of optimal
La Staying ahead \implies optimality.

· ·	· · ·		00	1 	•	Ą	- 	•	•	ł	h	9 L	Y	~ {		• • •	~ ~	•	•	•	•	• • • •	•	•	• •	· •	•	•	· · ·		•	•	•	· · ·	•	•	· · ·
• •	• •	• •	•	٠	0	• •	٠	٠	٠	÷	٠	٠	٠	٠	٠	• •	•	٠	÷	٠	•	• •	٠	÷	•	•	٠	÷	• •	•	٠	•	÷	• •	٠	•	• •
• •	• •	• •	•	•	•	• •	•	ľ	V	hc	ł	•	. (F	•	 	fh	L.	•		n bees	<u> </u>		λa	_d		•	•	• •		•	•	•	• •	•	•	• •
• •	• •		• •	•	•	· ·	•		2	(+	Fe	21	'e	v	, f	• •		P		ן פֿי∕		Hie	≤ S	•			•	•	••••	•	•	•	•	· ·	•	•	• •
			•	•	•		•		•		•	•	•	•	•	• •			•		•	• •	•	•	• •			•	• •		•	•				•	
• •	• •	• •	•	•	•		•	•	•		•	•		•	•	• •			•	•	•	• •	•	•	• •	•	•	•		•	•	•	•		•	÷	• •
• •	• •		•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•	٠	•	•	• •	•	•	•		•	•	• •	•	•	•	•	• •	•	•	
• •	• •		•	•	•	•••	•	•	•	•	•	•	•	•	•	• •		•	•		•	• •	•	•	• •	•	•	•	• •	•	•	•	•	• •	•	•	• •
• •	• •		•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •	•	•	• •		•	•	• •	•	•	•	•		•	e e	• •
• •	• •		•	•	•	• •	•	•	•		•	•	•	•	•	• •	•	•	•	•	•	• •	•	•		•	•	o	• •	•	•	•	•	• •	•	•	• •
• •	• •	• •	•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •	•	•	• •	•	•	•	• •		•	•	•	• •	•	•	• •
• •	• •			•			•	•	•		•	•	•	•	•			•	•	•	•	• •	•	•			•	•			•	•	•			•	• •
• •	• •		•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •	•	•	• •		•	•	• •	•	•	•	•		•	•	
• •	• •		•	•	•	• •	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	• •	•	•	• •	• •	•	•			•	•			•	e e	• •
• •	• •		•	•			•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	• •	•	•	• •		•	•	• •	•	•	•	•		•	•	• •
	• •		•											•		• •	•	٠								•				•		•					• •