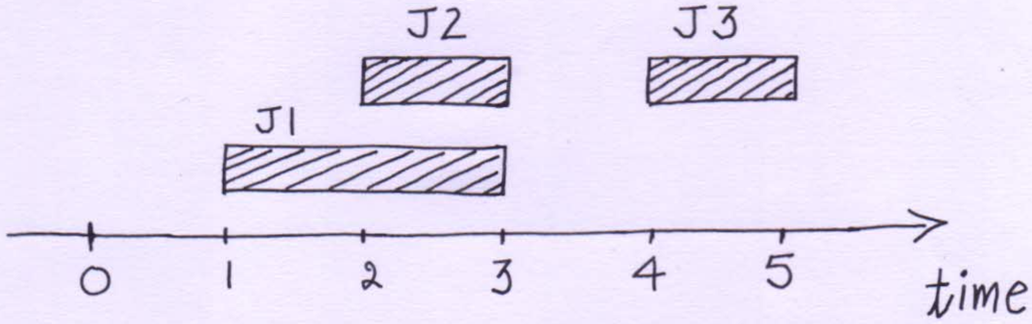
# JOB SCHEDULING

## BASED ON

## GREEDY APPROACH




$\{J_1, J_2\}$ OVERLAPPING

$\{J_2, J_3\}$ NON OVERLAPPING

$\{J_1, J_3\}$ NON OVERLAPPING.

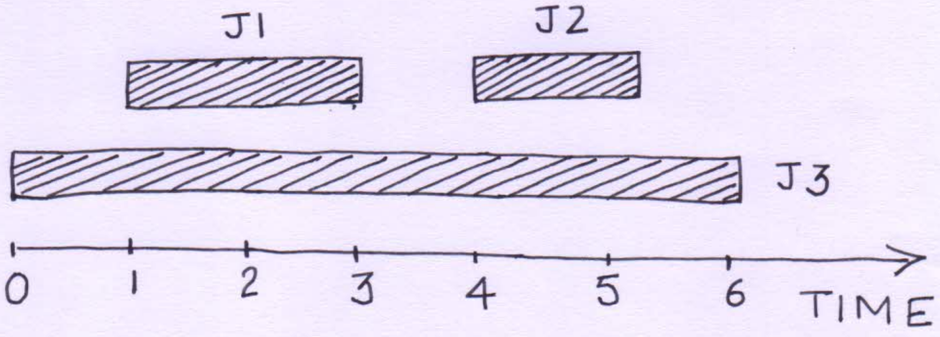
- $S = \{1, 2, 3, \ldots, n\}$ IS A SET OF $n$ JOBS/TASKS.
- EACH JOB $i \in S$ HAS A START TIME $S(i)$ and A FINISH TIME $F(i)$.
- SUBSET $R \subseteq S$ IS COMPATIBLE IF NO TWO TASKS IN $R$ ARE OVERLAPPING.
- GOAL: FIND A COMPATIBLE SUBSET AS LARGE AS POSSIBLE.

# GREEDY APPROACH - I

- ARRANGE THE JOBS IN A LIST L IN INCREASING ORDER OF THEIR START TIMES.

WHILE(L $\neq$ EMPTY)
REMOVE FIRST JOB FROM L.
IF (IT DOES NOT OVERLAP WITH ANY OF THE ALREADY SCHEDULED JOBS) THEN
SCHEDULE IT.
ELSE DISCARD IT
ENDIF
ENDWHILE.

CLAIM : GREEDY APPROACH-I IS SUB OPTIMAL.



# GREEDY APPROACH - 2

- ARRANGE THE JOBS IN A LIST L IN INCREASING ORDER OF THEIR DURATION.
- WHILE ( L $\neq$ EMPTY )

  REMOVE FIRST JOB FROM L

  IF ( IT DOES NOT OVERLAP WITH ANY OF THE ALREADY SCHEDULED JOBS ) THEN

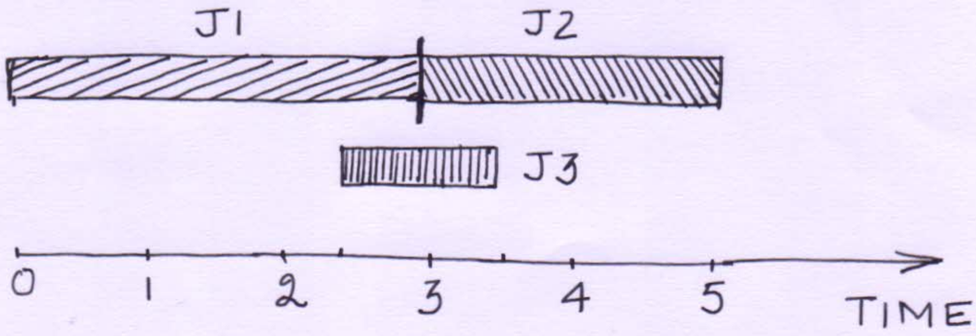
  SCHEDULE IT.

  ELSE DISCARD IT

  ENDIF

  ENDWHILE.

CLAIM : THE GREEDY APPROACH-2 IS SUB OPTIMAL.



# GREEDY APPROACH - 3

- ARRANGE THE JOBS IN A LIST L IN INCREASING ORDER OF NUMBER OF CONFLICTS
- WHILE ( L $\neq$ EMPTY )

  REMOVE FIRST JOB FROM L.

  IF ( IT DOES NOT OVERLAP WITH ANY OF THE ALREADY SCHEDULED JOBS ) THEN

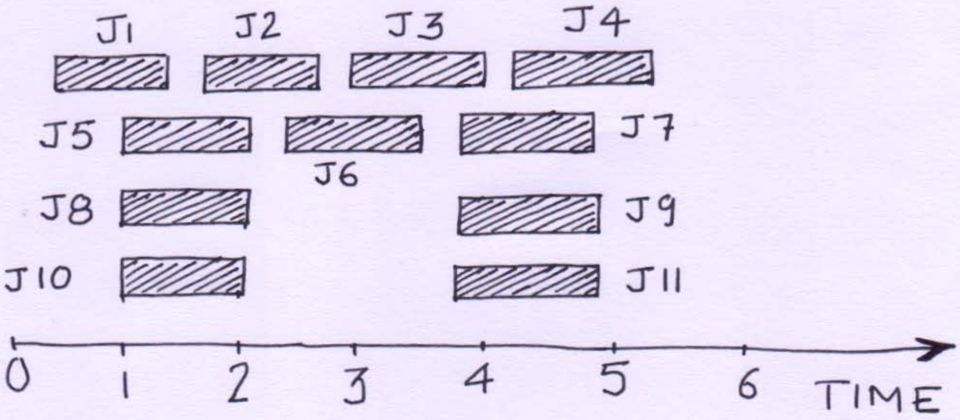
  SCHEDULE IT

  ELSE DISCARD IT

  END IF

  ENDWHILE.

CLAIM : THE GREEDY APPROACH-3 IS SUB OPTIMAL.



# GREEDY APPROACH - 4

- ARRANGE THE JOBS IN A LIST L IN INCREASING ORDER OF FINISH TIME

- WHILE( $L \neq$ EMPTY)

  REMOVE FIRST JOB FROM L

  IF( IT DOES NOT OVERLAP WITH ANY OF THE ALREADY SCHEDULED JOB) THEN

  SCHEDULE IT

  ELSE DISCARD IT

  END IF.

  END WHILE.

<u>CLAIM</u> : THE GREEDY APPROACH – 4 IS OPTIMAL.

<u>PROOF</u> : • SUPPOSE OUR APPROACH SELECTS K JOBS IN THE ORDER:
$i_1, i_2, i_3, \ldots, i_K$

• SUPPOSE THE OPTIMAL ALGO SELECTS $m$ JOBS IN THE ORDER:
$j_1, j_2, j_3, \ldots, j_m$

• TO PROVE : $K = m$

## OBSERVATION No. 1

$$F(i_1) \leq F(j_1)$$

## OBSERVATION No. 2

$$\forall r \leq k : F(i_r) \leq F(j_r)$$

## THEOREM :

IT IS THE CASE THAT $m = k$.

# EXTENSIONS

(1) ONLINE ALGORITHMS

(2) WEIGHTED JOBS SCHEDULING.