Aspetto greedy del problema:

$\exists$ attività di S_{ij} che sta sicuramente in A^*_{ij}

Qual'è questa attività?

Teorema: se $S_{ij} \neq \emptyset$, sia a_m l'attività tale che
$f_m = \min\{f_k : a_k \in S_{ij}\}$. allora

cioè quella che finisce prima di tutte

1) $\exists$ soluzione ottima A^*_{ij} tale che $a_m \in A^*_{ij}$

2) il sottoproblema $S_{im} = \emptyset$

$$\Rightarrow A^*_{ij} = \cancel{A^*_{im}} \cup \{a_m\} \cup A^*_{mj}$$

Dimostrazione:

2) banale; per assurdo suppongo $\exists\, a_k \in S_{im}$; allora
$$f_i \le s_k < \underbrace{f_k \le s_m < f_m}_{\text{assurdo!}}$$

1) cut & paste ("exchange argument"):

sia $\bar{A}_{ij}$ una soluzione ottima per S_{ij}

sia $a_k \in \bar{A}_{ij} : f_k = \min\{f_l : a_l \in \bar{A}_{ij}\}$

se $a_k = a_m$ ho finito

suppongo $a_k \neq a_m$; sia $A^*_{ij} = \bar{A}_{ij} \underset{\text{cut}}{\setminus} \{a_k\} \underset{\text{paste}}{\cup} \{a_m\}$

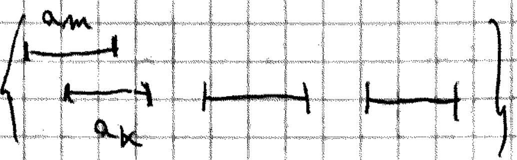
ora basta dimostrare che A^*_{ij} è soluzione ammissibile, visto che
$|A^*_{ij}| = |\bar{A}_{ij}|$

osserva che $f_m \leq f_k$ $\quad$ f_m è minimo globale

e che $f_k \leq s_\ell \quad \forall a_\ell \in \bar{A}_{ij}$

Quindi $f_m \leq f_k \leq s_\ell$, cioè a_m è compatibile con ogni a_ℓ:

$\bar{A}_{ij}$:



Abbiamo dimostrato la 1ª proprietà di scelta greedy del corso.

Quindi, cosa devo fare per risolvere S_{ij}:
1) scegliere a_m
2) risolvere S_{mj} $\quad$ (perché $A^*_{ij} = \{a_m\} \cup A^*_{mj}$)

Nota: approccio è top-down

Per risolvere $S = S_{0\,n+1}$ viene invocata la soluzione <u>solo di problemi del tipo $S_{m\,n+1}$</u> $\Rightarrow$ spazio dei sottoproblemi è ridotto da quadratico a <u>lineare</u>: al più $n+1$ sottoproblemi, cioè quelli che vanno da $m = 0$ a $m = n+1$

```
OPT (S_{i n+1})
    if S_{i n+1} ≠ ∅ then
        am : fm = min { fk : ak ∈ S_{i n+1} }      \\ scelta greedy
        return {am} ∪ OPT (S_{m n+1})
    else
        return ∅
```

"tail recursion", cioè la ricorsione appare solo in coda del codice

Implementazione dettagliata

REC_SEL (S, f, i) → vettore dei tempi di fine ordinati → dall'esterno la chiamo con $i=0$

```
REC_SEL (S, f, i)
    n ← length (S)
    m ← i + 1
    while [ (m ≤ n) and (S_m < f_i) ] do        → attività m ∉ S_{i,n+1}
        m ← m + 1
    if m ≤ n then return {a_m} ∪ REC_SEL (S, f, m)
    else return ∅
```

Modello di costo: confronti tra elementi di S e di f

Complessità: $\Theta(n)$ perché ogni attività viene toccata da un confronto una sola volta

↓

2 ordini di grandezza guadagnati con l'approccio greedy!

Codice iterativo:

```
GREEDY_SEL (S, f)
    n ← length (S)
    A ← {a_1}
    last ← 1                      // indica ultima attività selezionata
    for m = 2 to n do
        if s_m ≥ f_last then
            A ← A ∪ {a_m}
            last ← m
    return A
```