Algoritmi Greedy

↳ "incauto", "irruente"

- semplici, costruiscono l'ottimo per scelte successive
- problemi di ottimizzazione
- proprietà di sottostruttura ottima
- risolve un solo sottoproblema
- campo di applicazione limitato
- risolveremo problemi di selezione di attività e di compressione dati

Selezione delle attività compatibili

- risorsa condivisa
- insieme di attività $S = \{a_i : 1 \leq i \leq n\}$

  $a_i \rightarrow [s_i, f_i)$ $\quad 0 \leq s_i < f_i$

a_i e a_j sono compatibili $\iff [s_i, f_i) \cap [s_j, f_j) = \emptyset$

$\Updownarrow$

$f_i \leq s_j$ oppure $f_j \leq s_i$

esempio: aula P200

a_1 : $[10{:}30, 12{:}00)$

a_2 : $[11{:}30, 13{:}00)$

a_3 : $[12{:}30, 14{:}00)$

a_1 non è compatibile con a_2

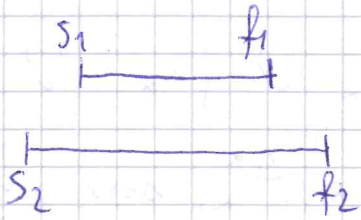
a_1 è compatibile con a_3

Problema: determinare un sottoinsieme di massima cardinalità di attività mutuamente (cioè, a coppie) compatibili.

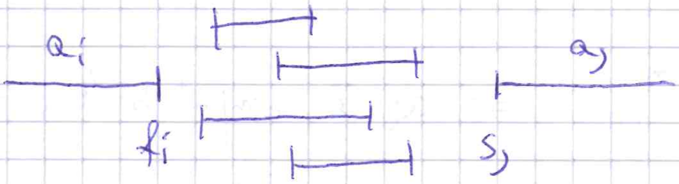
Proviamo con la programmazione dinamica:

ulteriore assunzione: $0 < f_1 \leq f_2 \leq \dots \leq f_n$

esempio:



Sottoproblemi: $S_{ij} = \{ a_k : f_i \leq s_k < f_k \leq s_j \}$



Osservazioni su S_{ij}:

1) $i > j \Rightarrow S_{ij} = \emptyset$

2) ci sono tutte le attività di indice k con $i < k < j$ in S_{ij}? No

3) $|S_{ij}| \leq j - i - 1$

Def.: $f_0 = 0$, $s_{n+1} = +\infty$

$\Rightarrow S = S_{0,n+1}$

Proprietà di sottostruttura ottima:

Sia A^*_{ij} un sottoinsieme di attività compatibili di S_{ij} di cardinalità massima (cioè, soluzione ottima di S_{ij})

- $S_{ij} = \emptyset \Rightarrow A^*_{ij} = \emptyset$ banale
- $S_{ij} \neq \emptyset \Rightarrow$ se $a_k \in A^*_{ij}$, allora

$$A^*_{ij} = A^*_{ik} \cup \{a_k\} \cup A^*_{kj}$$

dove A^*_{ik} e A^*_{kj} sono soluzioni ottime per S_{ik} e S_{kj}

dimostrazione:

$a_l, l \neq k, \in A^*_{ij} \Rightarrow a_l$ è compatibile con a_k, cioè

$f_i \leq s_l < f_l \leq s_k \Rightarrow a_l \in S_{ik}$

oppure

$s_k < f_k \leq s_l < f_l \leq s_j \Rightarrow a_l \in S_{kj}$

$$\Rightarrow A^*_{ij} = \underset{\text{cut}}{\cancel{\bar{A}_{ik}}} \cup \{a_k\} \cup \bar{A}_{kj}$$

paste A^*_{ik} → ottengo un insieme più grande di A^*_{ij}: impossibile

Ricorrenza:

$$c(i,j) = |A^*_{ij}|$$

$$c(i,j) = \begin{cases} 0 & S_{ij} = \emptyset \\ \max\limits_{a_k \in S_{ij}} \{c(i,k) + c(k,j) + 1\} & S_{ij} \neq \emptyset \end{cases}$$

→ complessità $\Theta(n^3)$