

Elementi strategie greedy

Un algoritmo greedy opera facendo ad ogni bivio una scelta ottimale al momento

Queste strategie non sempre producono una soluzione ottimale

Come sviluppare un algoritmo greedy

1. Determinare la sottostruttura ottimale del problema
2. Sviluppare una soluzione ricorsiva
3. Mostrare che facendo una scelta greedy rimane solo un sottoproblema da risolvere
4. Dimostrare che è sempre sicuro fare scelte greedy
5. Sviluppare un algoritmo ricorsivo che implementi le scelte greedy
6. Convertire l'algoritmo ricorsivo in un algoritmo iterativo

Questi passi mostrano la programmazione dinamica sottostante, quindi dimostrando l'applicabilità della programmazione dinamica siamo a metà dell'opera

Le strategie greedy non funzionano sempre. Ma ci sono due ingredienti chiave per il suo funzionamento

- Proprietà di sottostruttura ottimale
- Proprietà di scelte greedy

PROPRIETÀ DI SCELTA GREEDY

Un problema gode delle proprietà di scelte greedy se possiamo creare una soluzione ottimale globale

se forniamo questa una soluzione ottima greedy,
portando le scelte ottime locali

La programmazione dinamica risolve i sottoproblemi
prima di fare delle scelte

INVECE

La strategia greedy fa una scelta prima di risolvere
qualsunque sottoproblema

COME DIMOSTRIAMO LA PROPRIETÀ DI SCELTA OTTIMA

In genere si esamina una soluzione ottima e
si modifica facendo le scelte greedy, ottenendo
così una soluzione simile, ma non peggiore

DIMOSTRAZIONE ACTIVITY SELECTION

Assumendo che le attività siano ordinate per finish
time crescente, ovvero $f_1 \leq f_2 \leq \dots \leq f_n$

Sia A_k una soluzione ottima al sottoproblema S_k
ovvero A_k è una serie di attività compatibili di S_k
di cardinalità massima

Sia a_j l'attività A_k con il minore finish-time

Se $a_j = a_n$, abbiamo finito perché $a_n \in A_k$
soluzione ottima di S_k

Se $e_j \neq e_m$, vie $A'_k = (A_k - \{e_j\}) \cup \{e_m\}$

- $|A'_k| = |A_k|$
- Le attività di A'_k sono compatibili perché le attività in $A_k - \{e_j\}$ sono compatibili.
Inoltre $f_m \leq f_j$ ovvero f_m termine prima

Abbiamo dimostrato la scelta greedy per activity selector