

ALGORITMI E STRUTTURE DATI

Prof. Manuela Montangero

A.A. 2022/23

Algoritmi GREEDY

"E' vietata la copia e la riproduzione dei contenuti e immagini in qualsiasi forma.

E' inoltre vietata la redistribuzione e la pubblicazione dei contenuti e immagini non autorizzata espressamente dall'autore o dall'Università di Modena e Reggio Emilia."



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Classificazione di problemi

I problemi possono essere classificati per
tipo di soluzione cercata

- **PROBLEMI DECISIONALI**: la risposta al problema è **SI** o **NO** (oppure vero o falso) a seconda che l'INPUT del problema soddisfi o meno una certa proprietà.

Esempi: ricerca binaria decisionale, test di connettività di un grafo...

- **PROBLEMI DI RICERCA**: tra tutte le soluzioni possibili se ne cerca una (detta *soluzione ammissibile*) che soddisfa una certa condizione.

Esempi: ricerca binaria, ordinare in una sequenza in di numeri in ordine crescente, determinare un albero di copertura per una visita DFS in-order...

- **PROBLEMI DI OTTIMIZZAZIONE**: ad ogni soluzione ammissibile è associato un costo e si cerca la *soluzione ottima*, ovvero una soluzione ammissibile di costo minimo o massimo.

funzione obiettivo

Esempio: determinare i cammini minimi da sorgente singola

Problemi di ottimizzazione

- **PROBLEMI DI OTTIMIZZAZIONE:** ad ogni soluzione ammissibile è associato un costo e si cerca la *soluzione ottima*, ovvero una soluzione ammissibile di costo minimo o massimo.

Esempio: determinare i cammini minimi da sorgente singola

INPUT: Grafo indiretto connesso, un nodo sorgente e una funzione di peso sugli archi

OUTPUT: cammini minimi della sorgente ad ogni altro nodo del grafo

Soluzione ammissibile: Insieme di $V-1$ cammini, ognuno dei quali è il cammino di un nodo del grafo dalla sorgente

Costo soluzione ammissibile: insieme di $V-1$ valori, ognuno dei quali è il costo di uno dei cammini nella soluzione ammissibile

Funzione obiettivo: minimo

Soluzione ottima: Insieme di $V-1$ cammini, ognuno dei quali è il cammino di costo minimo di un nodo del grafo dalla sorgente

Problemi di ottimizzazione

Alcune tecniche per affrontare problemi di ottimizzazione

Algoritmi **GREEDY** (Golosi o Ingordi):

costituisce la soluzione in maniera incrementale, effettuando una serie di scelte in cui, ad ogni passo, si seleziona una parte della soluzione che permette di ottimizzare il costo della soluzione parziale costruita fino a quel momento.

Algoritmi di **PROGRAMMAZIONE DINAMICA**:
ne parleremo più avanti

Algoritmi GREEDY

Si applicano a problemi di ottimizzazione in cui devono essere selezionati degli oggetti che verificano una certa proprietà in modo da ottimizzare il costo associato agli oggetti selezionati

Approccio generale:

- Ordinare gli oggetti per "appetibilità" a far parte dalla soluzione ottima
- Generare la soluzione in maniera incrementale, **scegliendo iterativamente l'oggetto più appetibile tra quelli non ancora selezionati** —> **SCELTA GREEDY**

Ad ogni iterazione la scelta greedy fornisce la soluzione migliore in quel momento

Le scelte greedy non vengono mai modificate successivamente

Dopo ogni iterazione:

- La soluzione parziale calcolata "è più grande" che all'inizio dell'iterazione
- Il resto della soluzione è calcolato su un sottoproblema più piccolo

Algoritmi GREEDY

ESEMPIO

PROBLEMA:

INPUT: tagli di monete da 50, 10, 5, 1 cent e un intero resto

OUTPUT: selezione delle monete (eventualmente con ripetizione) che sommino a resto e siano in minor numero possibile

soluzione ammissibile: selezione di monete che sommino a resto

costo soluzione: numero di monete nella selezione

funzione obiettivo: minimo

soluzione ottima: soluzione ammissibile ottima

Algoritmi GREEDY

ESEMPIO

PROBLEMA:

INPUT: tagli di monete da 50, 10, 5, 1 cent e un intero resto

OUTPUT: selezione delle monete (eventualmente con ripetizione) che sommino a resto e siano in minor numero possibile

resto = 72

soluzione: $50 + 10 + 10 + 1 + 1$

Abbiamo adottato un approccio GREEDY

funziona sempre?

Con questo taglio di monete si

Algoritmi GREEDY

ESEMPIO

PROBLEMA:

INPUT: tagli di monete da 11, 5, 1 cent e un intero resto

OUTPUT: selezione delle monete (eventualmente con ripetizione) che sommino a resto e siano in minor numero possibile

resto = 15

soluzione con approccio greedy: $11 + 1 + 1 + 1 + 1$

soluzione ottima: $5 + 5 + 5$

con questo taglio di monete no

Algoritmi GREEDY

La tecnica **GREEDY** può essere applicata con successo se valgono le seguenti proprietà per il **PROBLEMA** e per **l'ALGORITMO**:

- Il **PROBLEMA** ha una **SOTTOSTUTTURA OTTIMA**:
una soluzione ottima al problema contiene al suo interno le soluzioni ottime dei sottoproblemi
Esempio: se un nodo v si trova sul cammino minimo P da s a u , allora la parte del cammino P da s a v è minimo per v
- **SCELTA GREEDY**:
la scelta greedy permette di scegliere un elemento della soluzione che fa parte della soluzione ottima
Esempio: la scelta greedy di Dijkstra è scegliere, ad ogni iterazione, il nodo nella coda con priorità con massima priorità