ASD Laboratorio 07

Cristian Consonni/Marta Fornasier

UniTN

2019-03-04

CONTATTI

ISTRUTTORI

Cristian Consonni (`cristian.consonni@unitn.it`)
Marta Fornasier (`marta.fornasier@studenti.unitn.it`)

RICEVIMENTO

Consonni: via email e ufficio Open Space 9, Povo 2 (dopo il ponte, di fronte all'ufficio del prof. Montresor)
Fornasier: via email

Calendario

04/03	Programmazione dinamica
06/03	Programmazione dinamica
10/04	Ricevimento (facoltativo)
15/05	Algoritmi approssimati
22/05	Progetto alg approssimati
29/05	Progetto alg approssimati

Progetto algoritmi approssimati

- Algoritmi approssimati (ultima parte del corso);
- Assumiamo gli stessi gruppi del primo semestre, in caso di cambiamenti, avvisare **entro il 15/05**;

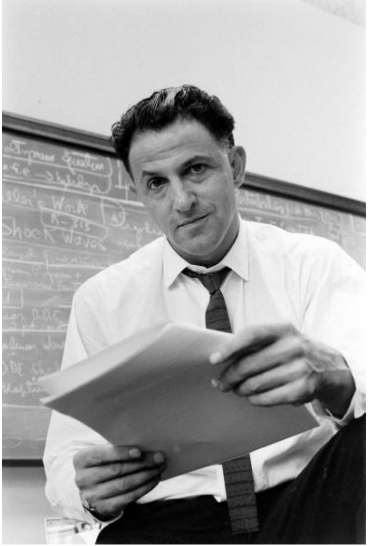
Programmazione dinamica

Definito da Richard Bellman negli anni '50, mentre lavorava con l'aeronautica militare statunitense

Inventò il termine per nascondere al suo capo la natura matematica della ricerca che faceva

Programmazione = Pianificazione (logistica)

Dinamica = Multilivello



Testo

Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Significato

Decidiamo il passo corrente in base allo stato corrente, senza riguardare le decisioni prese in precedenza.

Procedura tipica

Il tipico ragionamento è il seguente:

1. Definire i sottoproblemi/stati
2. Trovare l'equazione di ricorrenza che li collega
3. Utilizzare la memoization
4. Ridefinire l'algoritmo in maniera iterativa (opzionale)

Problema dello zaino

Dato uno zaino di capacità C, ed un insieme di N elementi con peso P_i vogliamo riempire al massimo lo zaino.

Sottoproblemi

Un sottoproblema è la conseguenza di una serie di decisioni precedenti.

Proprietà dei sottoproblemi

- Devono essere autosufficienti: lo stato del sottoproblema deve contenere abbastanza informazioni da permettere la costruzione di una strategia ottima
- Devono essere meno possibili

Esempi di sottoproblemi

- Sottoarray da i a N
- Sottoarray da i a j
- Sottoalbero radicato in n

Nel problema dello zaino: Riempire capacità C utilizzando oggetti da i in poi.

Equazione di ricorrenza

Ridefinire il costo della soluzione del problema corrente in base alle soluzioni di altri sottoproblemi.

Proprietà dell'equazione

- Esiste un ordinamento sui sottoproblemi, per evitare cicli infiniti
- Individuiamo possibili scelte
- Proviamo tutte e prendiamo la migliore
- Attenti ai casi base

Nel problema dello zaino: Possiamo inserire l'elemento i o non inserirlo.

$$S(C,i) = \begin{cases} Max \begin{cases} P_i + S(C - P_i, i+1) \\ S(C, i+1) \end{cases} & \text{if } i < N \\ -\inf & \text{if } C < 0 \\ 0 & \text{if } i == N \end{cases} \tag{1}$$

Memoization

Evitiamo di ricalcolare più volte la stessa funzione.

Condizione necessaria

La funzione di ricorrenza deve dipendere solo da:

- Strutture costanti nell'esecuzione (es: la lista dei pesi)
- I parametri della funzione (*i* e *C*)

Per ogni stato il cuore della funzione verrà eseguito soltanto una volta

Algorithm 1: `solve(State s):`

CASI BASE;
if $T[s] \neq null$ **then**
  return $T[s]$;
end
CHIAMATERICORSIVE;
$T[s] = sol$;
return $T[s]$;

Note di programmazione

Tabella di memoization

Matrice statica: `int T[MAXN][MAXM];`
Vettore dinamico:
`T=vector<vector<int>>(N,vector<int>(M,-1));`
Map: `map<pair<int,int>,int> T;`

State attenti alla memoria usata! Con la map spendete solo quello che utilizzate, ma pagate $O(\log n)$ per accesso.

Problemi

Testi completi su judge

Zaino

Dato uno zaino di capacità C ed un insieme di elementi di peso P_i e di valore V_i, trovare il massimo valore che è possible trasportare.

Sottoinsieme crescente di somma massima

Vi viene dato un array di interi. Per ogni elemento potete scegliere se includerlo nell'insieme soluzione, o se ignorarlo. Se un elemento X fa parte dell'insieme, tutti gli elementi che lo succedono nell'array e che hanno valore minore di X non possono essere inclusi nell'insieme. Bisogna massimizzare la somma dell'insieme.

PILLOLE

La zia ha N pillole in una bottiglia. Ogni giorno, pesca una pillola dalla bottiglia:

1. se è intera, la spezza in due metà, ne mangia una metà e rimette il resto nella bottiglia. (Caso I)
2. se è mezza pillola, la mangia (Caso M)

Con 3 pillole, sono possibili queste sequenze:
IIIMMM IIMIMM IIMMIM IMIIMM IMIMIM
Quante sequenze sono possibili con N pillole?