

Ottimizza la somma

Supponete di avere in input un vettore di n interi positivi distinti $V[1 \dots n]$ e un valore W . Scrivere un algoritmo che:

- 1 restituisca il massimo valore $X = \sum_{i=1}^n x[i]V[i]$ tale che $X \leq W$ e ogni $x[i]$ è un intero non negativo;
- 2 stampi il vettore x .

Ad esempio, per $V[] = \{18, 3, 21, 9, 12, 24\}$ e $W = 17$, una possibile soluzione ottima è $X[] = \{0, 2, 0, 1, 0, 0\}$ da cui deriva $X = 15$.

Discutere correttezza e complessità.

Hateville

Hateville è un villaggio particolare, composto da n case, numerate da 1 a n lungo una singola strada. Ad Hateville ognuno odia i propri vicini della porta accanto, da entrambi i lati; quindi il vicino i odia i vicini $i - 1$ e $i + 1$ (se esistenti). Hateville vuole organizzare una sagra e ha lanciato una raccolta fondi che è vostro compito organizzare. Ogni abitante i è in grado di donare una quantità $D[i]$, ma non intende partecipare ad una raccolta fondi a cui partecipano uno o entrambi i propri vicini.

Il vostro compito è il seguente:

- calcolare la quantità massima di fondi che può essere raccolta
- stampare gli indici delle case che dovranno donare

Illustrare il funzionamento e discutere la complessità degli algoritmi proposti.

Mosse su scacchiera

Supponete di avere una scacchiera $n \times n$ e un pedone che dovete muovere dall'estremità inferiore a quella superiore. Un pedone si può muovere (1) una casella in alto, oppure (2) una casella in diagonale alto-destra, oppure (3) una casella in diagonale alto-sinistra. Non può tornare indietro. Quando una cella (x, y) viene visitata, guadagnate un valore reale $p(x, y)$.

Calcolare un percorso da una qualunque casella dell'estremità inferiore ad una qualunque casella dell'estremità superiore, massimizzando il profitto.

6	7	4	7	<u>8</u>
7	6	1	1	<u>4</u>
3	5	7	<u>8</u>	2
2	6	<u>7</u>	0	2
7	3	5	<u>6</u>	1

Quadrato binario

Sia $A[1 \dots n, 1 \dots n]$ una matrice di valori booleani 0/1. Scrivere un algoritmo che restituisce la dimensione del più grande quadrato composto da valori 1. Ad esempio, nella matrice seguente, i quadrati di dimensione massima (ve ne sono due, di cui uno evidenziato in grassetto) hanno dimensione pari a 4.

1	0	1	0	1	0	0
1	0	1	1	1	1	0
0	1	1	1	1	1	0
0	0	1	1	1	1	0
1	1	1	1	1	1	0
1	1	1	1	1	1	0

Ottimizza la somma

$$X[i, w] = \begin{cases} -\infty & i \geq 0 \wedge w < 0 \\ 0 & i = 0 \vee w = 0 \\ \max\{X[i-1, w], X[i, w - V[i]] + V[i]\} & \text{altrimenti} \end{cases}$$

Ottimizza la somma

```
integer sum(integer[] V, integer i, integer w, integer[][] X)
```

```
if  $w < 0$  then
```

```
   $\perp$  return  $-\infty$ 
```

```
if  $i = 0$  or  $w = 0$  then
```

```
   $\perp$  return 0
```

```
if  $X[i, w] = \perp$  then
```

```
   $X[i, w] \leftarrow \max\{\text{sum}(X, i - 1, w), \text{sum}(X, i, w - V[i]) + V[i]\}$ 
```

```
return  $X[i, w]$ 
```

$$M[i] = \begin{cases} 0 & i = 0 \\ D[1] & i = 1 \\ \max\{M[i-1], M[i-2] + D[i]\} & i > 2 \end{cases}$$

Hateville

```
integer fundraising(integer[]  $D$ , integer  $n$ )
```

```
integer[]  $M \leftarrow$  new integer[1... $n$ ]           % Calcola il vettore  $M$   
 $M[0] \leftarrow 0$   
 $M[1] \leftarrow D[1]$   
for  $i \leftarrow 2$  to  $n$  do  
  |  $M[i] \leftarrow \max(D[i-1], D[i-2] + D[i])$   
  
integer  $i \leftarrow n$ ;                               % Stampa gli indici selezionati  
while  $i > 2$  do  
  | if  $M[i] = M[i-2] + D[i]$  then  
  |   | print  $i$   
  |   |  $i \leftarrow i - 2$   
  | else  
  |   |  $i \leftarrow i - 1$   
  |  
  
if  $i > 0$  then  
  | print  $i$   
  
return  $M[n]$                                          % Ritorna la quantità massima raccoglibile
```

Quadrato binario

$$M[i, j] = \begin{cases} 0 & A[i, j] = \mathbf{false} \\ 1 & A[i, j] = \mathbf{true} \wedge i = n \vee j = n \\ \min\{A[i + 1, j], \\ \quad A[i + 1, j + 1], \\ \quad A[i, j + 1]\} + 1 & \text{altrimenti} \end{cases}$$

Mosse su scacchiera

$$g[x, y] = \begin{cases} -\infty & x < 1 \vee x > n \\ p(x, y) & y = n \\ p(x, y) + \max_{d \in \{-1, 0, +1\}} \{ g[x + d, y + 1] \} & \text{otherwise} \end{cases}$$

Mosse su scacchiera

search-path(integer[][] p , integer n)

{ Calcola la tabella g }

for $x \leftarrow 1$ to n do $g[x, n] \leftarrow p[x, n]$

for $y \leftarrow n - 1$ downto 1 do

 for $x \leftarrow 1$ to n do

$g[x, y] \leftarrow -\infty$

 foreach $d \in \{-1, 0, +1\}$ do

 integer $x' \leftarrow x + d$

 if $x' \geq 1$ and $x' \leq n$ then

 integer $t \leftarrow g[x', y + 1] + p[x, y]$

 if $t > g[x, y]$ then

$g[x, y] \leftarrow t$

$m[x, y] \leftarrow d$

Quadrato binario

{ Cerca la casella iniziale con massimo guadagno }

integer x

for $i \leftarrow 1$ **to** n **do**

if $g[i, 1] > g[x, 1]$ **then**
 $x \leftarrow i$

{ Stampa il percorso }

for $y \leftarrow 1$ **to** $n - 1$ **do**

print “ $(x, y) \leftarrow (x + m[x, y], y + 1)$ ”
 $x \leftarrow x + m[x, y]$
