

Secondo allenamento

Olimpiadi Italiane di Informatica - Selezione territoriale

Luca Chiodini

luca@chiodini.org - l.chiodini@campus.unimib.it

15 marzo 2016

1. Lettura di un problema tratto dalle OIS
2. Soluzione naïve
3. Spiegazione teorica
4. Soluzione ottima

Lettura di un problema tratto
dalle OIS

“Rifornimenti ai distributori” (prima gara OIS 2015)

“Rifornimenti ai distributori” (prima gara OIS 2015)

Gabriele ha appena preso la patente, e decide di invitare tutti i suoi amici a fare una gita. Dato che il viaggio è lungo ben K chilometri, sa che forse dovrà fermarsi a fare il pieno di benzina: a tal proposito Gabriele ha segnato a che distanza dalla partenza ci sono gli N distributori che si trovano lungo il tragitto. Sapendo che la sua macchina fa al massimo M chilometri con un pieno, e che alla partenza ha già il serbatoio pieno, aiuta Gabriele a pianificare i rifornimenti di modo da fare benzina il minor numero possibile di volte.

“Rifornimenti ai distributori” (prima gara OIS 2015)

Il file `input.txt` è composto da due righe. La prima riga contiene i tre interi N, M, K separati da uno spazio. La seconda riga contiene N interi separati da uno spazio, le distanze D_i dei distributori in ordine crescente.

<code>input.txt</code>	<code>output.txt</code>
5 50 100	1
29 35 50 77 83	

“Rifornimenti ai distributori” (prima gara OIS 2015)

Il file `input.txt` è composto da due righe. La prima riga contiene i tre interi N, M, K separati da uno spazio. La seconda riga contiene N interi separati da uno spazio, le distanze D_i dei distributori in ordine crescente.

input.txt	output.txt
5 50 100	1
29 35 50 77 83	

input.txt	output.txt
10 30 100	6
1 31 33 38 62 69 93 97 98 99	

Soluzione naïve

Esempio

È possibile che la risposta al problema sia 0? È possibile che non ci sia risposta al problema?

“Rifornimenti ai distributori” - Soluzione naïve

Esempio

È possibile che la risposta al problema sia 0? È possibile che non ci sia risposta al problema?

Assunzioni

- $1 \leq N \leq 100\,000$.
- $1 \leq M \leq K \leq 1\,000\,000$.
- $1 \leq D_i < D_{i+1} < K$ per ogni $i = 0 \dots N - 2$.
- È sempre possibile raggiungere la destinazione: la distanza tra due distributori successivi non supera mai M , il numero di chilometri che la macchina è in grado di percorrere con un pieno.

“Rifornimenti ai distributori” - Soluzione naïve

In questo genere di problemi, un approccio che funziona *sempre* è il cosiddetto metodo “forza bruta” (detto informalmente: “le provo tutte”). Come si può applicare in questo problema?

29	35	50	77	83
----	----	----	----	----

“Rifornimenti ai distributori” - Soluzione naïve

In questo genere di problemi, un approccio che funziona *sempre* è il cosiddetto metodo “forza bruta” (detto informalmente: “le provo tutte”). Come si può applicare in questo problema?

29	35	50	77	83
----	----	----	----	----

0	0	0	0	0
---	---	---	---	---

0	0	0	0	1
---	---	---	---	---

0	0	0	1	0
---	---	---	---	---

...

1	1	1	1	1
---	---	---	---	---

“Rifornimenti ai distributori” - Soluzione naïve

Dato un vettore V , dove $V_i = 1$ sse è previsto un rifornimento in quel distributore, siamo in grado di stabilire se con quei rifornimenti arriviamo in fondo?

$$M = 50 \quad K = 100$$

29	35	50	77	83
----	----	----	----	----

1	0	1	1	0
---	---	---	---	---

“Rifornimenti ai distributori” - Soluzione naïve

Dato un vettore V , dove $V_i = 1$ sse è previsto un rifornimento in quel distributore, siamo in grado di stabilire se con quei rifornimenti arriviamo in fondo?

$$M = 50 \quad K = 100$$

29	35	50	77	83
----	----	----	----	----

1	0	0	0	1
---	---	---	---	---

“Rifornimenti ai distributori” - Soluzione naïve

Dato un vettore V , dove $V_i = 1$ sse è previsto un rifornimento in quel distributore, siamo in grado di stabilire se con quei rifornimenti arriviamo in fondo?

$$M = 50 \quad K = 100$$

29	35	50	77	83
----	----	----	----	----

0	0	0	1	1
---	---	---	---	---

“Rifornimenti ai distributori” - Soluzione naïve

Dato un vettore V , dove $V_i = 1$ sse è previsto un rifornimento in quel distributore, siamo in grado di stabilire se con quei rifornimenti arriviamo in fondo?

$$M = 50 \quad K = 100$$

29	35	50	77	83
----	----	----	----	----

0	1	0	0	0
---	---	---	---	---

“Rifornimenti ai distributori” - Soluzione naïve

Soluzione

La soluzione consiste quindi nel cercare tra tutte le possibili configurazioni *valide* quella che usa il *minor numero* di distributori.

“Rifornimenti ai distributori” - Soluzione naïve

Soluzione

La soluzione consiste quindi nel cercare tra tutte le possibili configurazioni *valide* quella che usa il *minor numero* di distributori.

Complessità computazionale

Qual è la complessità computazionale della soluzione?

“Rifornimenti ai distributori” - Soluzione naïve

Soluzione

La soluzione consiste quindi nel cercare tra tutte le possibili configurazioni *valide* quella che usa il *minor numero* di distributori.

Complessità computazionale

Qual è la complessità computazionale della soluzione?

- Intuitivamente: abbiamo due possibilità per la prima posizione del vettore. Quindi, fissata la prima posizione, ne abbiamo altre due per la seconda posizione (e così via). In tutto quindi

$$\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_{N \text{ volte}}$$

“Rifornimenti ai distributori” - Soluzione naïve

Soluzione

La soluzione consiste quindi nel cercare tra tutte le possibili configurazioni *valide* quella che usa il *minor numero* di distributori.

Complessità computazionale

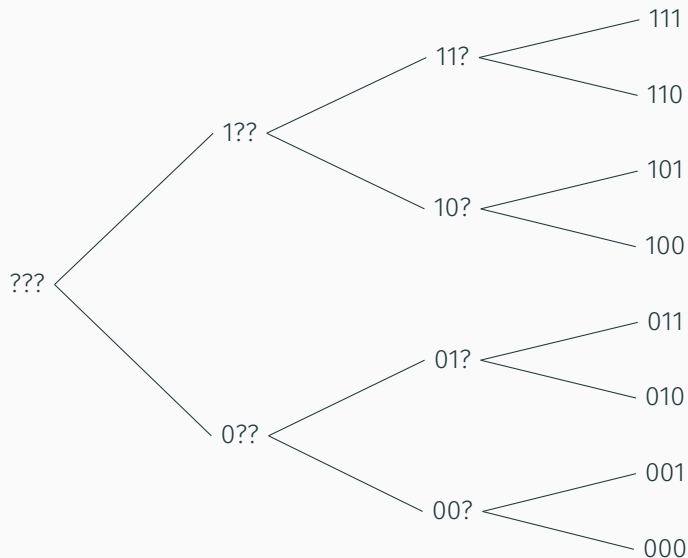
Qual è la complessità computazionale della soluzione?

- Intuitivamente: abbiamo due possibilità per la prima posizione del vettore. Quindi, fissata la prima posizione, ne abbiamo altre due per la seconda posizione (e così via). In tutto quindi

$$\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_{N \text{ volte}}$$

- Calcolo combinatorio: disposizioni con ripetizione di due elementi (0 - 1) su N posizioni $\implies 2^N$

“Rifornimenti ai distributori” - Soluzione naïve



“Rifornimenti ai distributori” - Soluzione naïve

```
int genera_disposizioni(bool V[], int posizione)
{
    if (posizione == N)
    {
        if (controlla(V))
            // Conta rifornimenti e aggiorna minimo.
        }
    else
    {
        V[posizione] = false;
        genera_disposizioni(V, posizione + 1);
        V[posizione] = true;
        genera_disposizioni(V, posizione + 1);
    }
}
```

Spiegazione teorica

Greedy

Alcuni problemi possono essere risolti facendo di volta in volta la scelta migliore localmente. Questa serie di scelte migliori locali, cioè dipendenti solo da quello che si sa al momento della scelta, si riflette alla fine nella soluzione migliore possibile.

Esempio: acquisto di biglietti per un concerto

Esistono M siti che vendono biglietti per lo stesso concerto a prezzi diversi. Devo comprare N biglietti (ovviamente spendendo il meno possibile). Cosa faccio?

Esempio: acquisto di biglietti per un concerto

Esistono M siti che vendono biglietti per lo stesso concerto a prezzi diversi. Devo comprare N biglietti (ovviamente spendendo il meno possibile). Cosa faccio? Seguo questo semplice algoritmo:

- Scelgo il sito dove i biglietti costano meno.
- Compro tutti i biglietti disponibili su quel sito per arrivare ad avere N biglietti
 - Se ora ho N biglietti, ho finito.
 - Se non ho ancora N biglietti, ripeto dall'inizio il procedimento.

Esempio: acquisto di biglietti per un concerto

- L'algoritmo è corretto?

Esempio: acquisto di biglietti per un concerto

- L'algoritmo è corretto? Sì, perché non c'è alcuna convenienza a scegliere i biglietti partendo da un sito in cui il costo non sia il minimo.
- È necessario applicare un algoritmo di ordinamento?

Esempio: acquisto di biglietti per un concerto

- L'algoritmo è corretto? Sì, perché non c'è alcuna convenienza a scegliere i biglietti partendo da un sito in cui il costo non sia il minimo.
- È necessario applicare un algoritmo di ordinamento? Sì (a meno che i dati non siano già ordinati)
- Qual è la complessità computazionale?

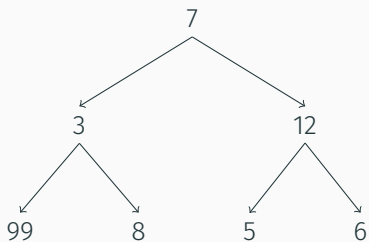
Esempio: acquisto di biglietti per un concerto

- L'algoritmo è corretto? Sì, perché non c'è alcuna convenienza a scegliere i biglietti partendo da un sito in cui il costo non sia il minimo.
- È necessario applicare un algoritmo di ordinamento? Sì (a meno che i dati non siano già ordinati)
- Qual è la complessità computazionale?
 - $O(M \log M + N)$ con ordinamento;
 - $O(N)$ se i dati sono già ordinati opportunamente.

Esempio di problema NON greedy

Esempio di problema NON greedy

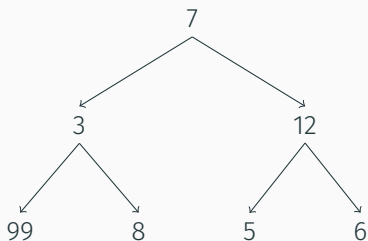
Determinare la somma massima ottenibile partendo dalla radice e arrivando a una foglia.



È un problema greedy?

Esempio di problema NON greedy

Determinare la somma massima ottenibile partendo dalla radice e arrivando a una foglia.



È un problema greedy? No, perché la miglior scelta locale non porta alla miglior scelta globale.

Soluzione ottima

Cosa si può osservare nell'esempio di input?

29	35	50	77	83
----	----	----	----	----

Cosa si può osservare nell'esempio di input?

29	35	50	77	83
----	----	----	----	----

I distributori sono ordinati. Il problema è greedy?

Dimostrazione

Consideriamo l' i -esimo distributore a cui arriviamo con il serbatoio che contiene ancora P litri di benzina (osserviamo che $P \leq M$). Consideriamo inoltre l' $i + 1$ -esimo distributore. Possono verificarsi due casi:

- La distanza tra i due distributori vale $D_{i+1} - D_i > P$. È imperativo fermarsi nell' i -esimo distributore, perché il successivo è troppo lontano.
- La distanza tra i due distributori vale $D_{i+1} - D_i \leq P$. Non c'è alcuna convenienza a fermarsi nell' i -esimo distributore, visto che si può raggiungere il successivo e fare il pieno.

Complessità computazionale

Qual è la complessità computazionale dell'algoritmo che abbiamo progettato?

Complessità computazionale

Qual è la complessità computazionale dell'algoritmo che abbiamo progettato? $O(N)$.

Complessità computazionale

Qual è la complessità computazionale dell'algoritmo che abbiamo progettato? $O(N)$.

Consiglio implementativo

Prestate attenzione all'ultimo distributore: non basta fermarsi lì, ma è necessario arrivare a percorrere *tutti* i K chilometri.

Esercizio

Implementare una soluzione per il problema “distributori”.

Riferimenti

Questa presentazione e soluzione in C++ dell’esercizio:

https://files.chiodini.org/OII_Territoriali_2016/

- Piattaforma di allenamento con correttore e forum:
<https://cms.di.unipi.it>
- Guida alle selezioni territoriali del prof. Bugatti:
http://www.imparando.net/sito/olimpiadi_di_informatica.htm
- Breve introduzione alle gare di programmazione:
https://files.chiodini.org/Introduzione_gare_programmazione.pdf