

Laboratorio di Algoritmi e Strutture Dati

Docente: S. Aguzzoli

Progetto “Compagnie aeree”

valido per l'appello di febbraio 2013

1 Il problema

Numerose *compagnie aeree* si spartiscono il traffico aereo internazionale che collega vari *aeroporti* sparsi in giro per il mondo. Le compagnie aeree e gli aeroporti sono identificati da stringhe di lunghezza arbitraria sull'alfabeto $\{a, b, c, \dots, A, B, C, \dots, 0, 1, 2, \dots, 9\}$. Indicheremo con \mathcal{C} l'insieme delle compagnie aeree e con \mathcal{A} l'insieme degli aeroporti. Ciascuna compagnia offre ai suoi clienti alcune *tratte* specifiche, che collegano un aeroporto di partenza con uno di arrivo. Ogni tratta è caratterizzata da una *lunghezza* in chilometri e dal *prezzo* in dollari con cui è messa in vendita alla clientela. Se la compagnia $C \in \mathcal{C}$ offre una tratta t che parte dall'aeroporto a e arriva all'aeroporto b , allora diciamo che a e b sono *serviti* da C . Se $C \in \mathcal{C}$ offre una tratta dall'aeroporto a all'aeroporto b , tale tratta può essere percorsa anche in senso inverso, e dunque collegare b ad a .

Esempio 1. *Supponiamo che le compagnie siano:*

$\mathcal{C} = \{ \text{Aliveloci, Boosterfly, Condorwings, Dumboplanes} \}$,

e che le tratte offerte siano:

- Aliveloci *offre:*
 - Anversa-Istanbul di 2500 km a 1500 \$;
 - Anversa-Helsinki di 1700 km a 1200 \$;
 - Anversa-Berlino di 800 km a 700 \$;
 - Berlino-Helsinki di 500 km a 600 \$;
 - Helsinki-Istanbul di 500 km a 500 \$;
 - Helsinki-Colonia di 900 km a 900 \$;
 - Colonia-Ginevra di 300 km a 200 \$;
 - Colonia-Istanbul di 1000 km a 600 \$;
 - Ginevra-Istanbul di 400 km a 200 \$;
- Boosterfly *offre:*
 - Anversa-Jerevan di 600 km a 600 \$;
 - Jerevan-Istanbul di 1500 km a 1100 \$;
 - Berlino-Colonia di 900 km a 1100 \$;
 - Edimburgo-Dublino di 300 km a 500 \$;
 - Edimburgo-Francoforte di 200 km a 100 \$;
 - Dublino-Francoforte di 300 km a 400 \$;
- Condorwings *offre:*
 - Jerevan-Istanbul di 1500 km a 1200 \$;

- Berlino-Anversa di 700 km a 800 \$;
- Berlino-Colonia di 2000 km a 1000 \$;
- Berlino-Dusseldorf di 3000 km a 2000 \$;
- Colonia-Dusseldorf di 2000 km a 900 \$;

• Dumboplanes offre:

- Jerevan-Berlino di 1000 km a 800 \$;
- Berlino-Ginevra di 1000 km a 800 \$;
- Ginevra-Francoforte di 900 km a 1200 \$.

Si noti che le stesse coppie di aeroporti possono essere servite da compagnie diverse. Ad esempio la tratta Anversa-Berlino è offerta sia da Aliveloci sia da Condorwings. Si noti anche che la tratta Anversa-Berlino offerta da Aliveloci è più lunga di quella offerta da Condorwings, ma costa di meno.

Non tutte le coppie di aeroporti sono collegate da una tratta; a volte il *viaggio* tra gli aeroporti di partenza e di arrivo è ottenuto combinando più tratte successive e prevede quindi degli *scali*, ovvero degli aeroporti di passaggio tra quelli di partenza e di arrivo. In questo caso la lunghezza e il prezzo del viaggio sono date dalla somma delle lunghezze e dei prezzi, rispettivamente, delle tratte che definiscono il *viaggio*. Formalmente, $v = (t_1, t_2, \dots, t_n)$ è un viaggio tra gli aeroporti $a, b \in \mathcal{A}$ se t_i è una tratta per ogni $i = 1, 2, \dots, n$, l'aeroporto di partenza di t_1 è a , l'aeroporto di arrivo di t_i coincide con l'aeroporto di partenza di t_{i+1} per ogni $i = 1, 2, \dots, n-1$, e l'aeroporto di arrivo di t_n è b .

Si dice che un *viaggio* è offerto da una certa compagnia aerea C se tutte le tratte che lo compongono sono offerte da C .

Esempio 2. Assumendo i dati dell'esempio ??, si noti che nessuna tratta connette Anversa e Dusseldorf, ma vi sono viaggi con scalo tra Anversa e Dusseldorf. Ad esempio il viaggio

(Condorwings:Anversa-Berlino, Condorwings:Berlino-Dusseldorf)

lungo 3700 km di costo 2800 \$; o il viaggio

(Aliveloci:Anversa-Berlino, Condorwings:Berlino-Dusseldorf)

lungo 3800 km di costo 2700 \$; o ancora il viaggio

(Aliveloci:Anversa-Berlino, Condorwings:Berlino-Colonia, Condorwings:Colonia-Dusseldorf)

lungo 4800 km di costo 2600 \$.

Si noti che il primo viaggio è più corto degli altri due, mentre il terzo è il più economico dei tre, anche se contempla due scali, mentre il primo e il secondo viaggio ne prevedono solo uno.

La compagnia Dumboplanes offre un viaggio da Francoforte a Jerevan:

(Dumboplanes:Francoforte-Ginevra, Dumboplanes:Ginevra-Berlino, Dumboplanes:Berlino-Jerevan).

Nessun'altra compagnia offre un viaggio da Francoforte a Jerevan.

Una compagnia aerea $C \in \mathcal{C}$ cerca di predisporre un *piano di voli*, vale a dire un insieme $P(C)$ di tratte tale che

1. $P(C)$ è *sufficiente*: ossia le tratte in $P(C)$ permettono di viaggiare da a a b per ogni coppia di aeroporti serviti da C ;
2. La somma delle lunghezze delle tratte in $P(C)$ è minima rispetto a tutti gli insiemi sufficienti di tratte.

Si osservi che non tutte le compagnie aeree dispongono necessariamente di un piano di voli, e che una compagnia può disporre di diversi piani di voli.

Esempio 3. La compagnia Aliveloci dispone di un piano di voli. Ad esempio il seguente è un piano di voli per Aliveloci: {Helsinki-Istanbul, Helsinki-Berlino, Berlino-Anversa, Istanbul-Ginevra, Ginevra-Colonia}.

La compagnia Dumboplanes dispone di un piano di voli. Ad esempio il seguente è un piano di voli per Dumboplanes: {Berlino-Jerevan, Berlino-Ginevra, Ginevra-Francoforte}.

La compagnia Boosterfly non dispone di un piano di voli. Infatti Boosterfly non offre viaggi fra Anversa e Colonia.

La compagnia Condorwings non dispone di un piano di voli. Infatti Condorwings non offre viaggi fra Istanbul e Dusseldorf.

Per fidelizzare i clienti, le compagnie offrono ai loro viaggiatori dei *jet-points*: per ogni tratta è definito un *valore in jet-points*; i viaggiatori possono *accumulare* jet-points nelle varie tratte e usarli, al posto dei dollari, per acquistare delle tratte; i jet-points di un viaggio si accumulano additivamente, cioè i punti acquisiti con una tratta si sommano ai punti precedentemente acquisiti. Le compagnie sono raggruppate in *partenariati*: i jet-points acquisiti con una compagnia aerea sono utilizzabili per comprare biglietti aerei anche di altre compagnie facenti parte dello stesso partenariato. Indicheremo con \mathcal{P} l'insieme dei partenariati.

Esempio 4. Supponiamo che $\mathcal{P} = \{\{\text{Aliveloci}\}, \{\text{Boosterwings}\}, \{\text{Condorwings}, \text{Dumboplanes}\}\}$ sia l'insieme dei partenariati.

Si supponga inoltre che i jet-points acquisibili nelle varie tratte siano i seguenti:

- *Tratte Aliveloci:*
 - Anversa-Istanbul 100 *jet points*
 - Anversa-Helsinki 140 *jet points*
 - Anversa-Berlino 70 *jet points*
 - Berlino-Helsinki 250 *jet points*
 - Helsinki-Istanbul 100 *jet points*
 - Helsinki-Colonia 200 *jet points*
 - Colonia-Ginevra 200 *jet points*
 - Colonia-Istanbul 300 *jet points*
 - Ginevra-Istanbul 300 *jet points*
- *Tratte Boosterfly:*
 - Anversa-Jerevan 500 *jet points*
 - Jerevan-Istanbul 400 *jet points*
 - Berlino-Colonia 300 *jet points*
 - Edimburgo-Dublino 200 *jet points*
 - Edimburgo-Francoforte 200 *jet points*
 - Dublino-Francoforte 300 *jet points*
- *Tratte Condorwings:*
 - Jerevan-Istanbul 500 *jet points*
 - Berlino-Anversa 400 *jet points*
 - Berlino-Colonia 300 *jet points*
 - Berlino-Dusseldorf 600 *jet points*

- Colonia-Dusseldorf 200 *jet points*
- *Tratte Dumboplanes:*
 - Jerevan-Berlino 300 *jet points*
 - Berlino-Ginevra 250 *jet points*
 - Ginevra-Francoforte 400 *jet points*

Si noti che la tratta Anversa-Berlino conferisce 70 *jet points* se effettuata con Aliveloci, mentre ne conferisce 400 se effettuata con Condorwings.

Il viaggio (Dumboplanes:Berlino-Jerevan, Condorwings:Jerevan-Istanbul), effettuato con il partenariato {Condorwings, Dumboplanes}, costa 2000 \$ e conferisce 800 *jet points*.

Il viaggio (Aliveloci:Berlino-Helsinki, Aliveloci:Helsinki-Istanbul), effettuato con il partenariato {Aliveloci}, costa 1100 \$ e conferisce 350 *jet points*.

Il programma di fidelizzazione per la compagnia Swindle-fly $\in \mathcal{A}$ è scaduto a fine anno e i *jet-points* non spesi non saranno più utilizzabili per comprare tratte aeree. La compagnia consente però *generosamente* ai suoi clienti di usare i propri *jet-points* rimasti per acquistare dei buoni in dollari chiamati *swindles*; questi *swindles* potranno essere usati nel 2013 sull'acquisto di tratte offerte dalla compagnia Swindle-fly.

Esempio 5. Si supponga che il costo in *jet points* degli *swindles* sia come riportato nella tabella seguente:

<i>Swindles (valore in dollari)</i>	<i>Jet-points</i>
100	20.000
107	22.000
124	24.000
133	26.000
139	28.000
155	30.000
172	32.000
178	34.000
184	36.000
190	38.000
195	40.000

Se il totale di *jet-points* è 60.000 allora il valore massimo in dollari che può essere realizzato acquistando *swindles* è pari a 311 \$ (uno *swindle* da 133 \$ e uno *swindle* da 178 \$). Se il totale di *jet-points* è 100.000 allora il valore massimo in dollari che può essere realizzato acquistando *swindles* è pari a 528 \$ (due *swindle* da 172 \$ e uno *swindle* da 184 \$). Se il totale di *jet-points* è 140.000 allora il valore massimo in dollari che può essere realizzato acquistando *swindles* è pari a 740 \$ (tre *swindle* da 172 \$, uno *swindle* da 124 \$ e uno *swindle* da 100 \$).

2 Specifiche di progettazione

Si deve progettare un programma che sia in grado di gestire i dati relativi alle compagnie aeree e alle tratte da loro offerte e di eseguire le seguenti operazioni. Il progetto del programma va illustrato in una relazione scritta.

La progettazione deve prevedere la scelta delle strutture dati da usare per rappresentare i dati e gli algoritmi da applicare per risolvere in maniera efficiente i problemi descritti nella traccia. Non basta limitarsi a riferimenti generici alle tecniche algoritmiche utilizzate (es: “l’operazione X si risolve con un algoritmo greedy”) ma è necessario dettagliare le procedure da utilizzare, tramite pseudocodice o direttamente il linguaggio C.

In particolare si richiede di analizzare, in funzione delle scelte di progettazione fatte, quale risulta essere il costo delle diverse operazioni richieste dalla specifica.

Si richiede inoltre di fornire una rassegna *esauriente* di esempi che potrebbero essere usati per testare il programma e che mettono in evidenza particolari caratteristiche del suo funzionamento (non solo casi tipici di input, ma anche casi limite e/o situazioni patologiche; input che evidenzino la differenza di prestazioni tra le soluzioni progettuali scelte e altre più semplicistiche).

Non si richiede un'implementazione completa del progetto; è sufficiente fornire l'ossatura del programma (eventualmente suddiviso su più file) contenente in particolare: le definizioni dei tipi fondamentali, i prototipi delle funzioni che realizzano le operazioni specificate nella traccia, e tutte le porzioni di codice utili ad illustrarne il loro uso.

Facoltativamente, è possibile consegnare un'implementazione completa e funzionante del progetto.

2.1 Operazioni da eseguire

Si noti che le operazioni richieste sono liberamente implementabili; in particolare, non vanno necessariamente intese come prototipi di funzioni.

- **apertura_aeroporto**(*nome*)

Aggiunge ad \mathcal{A} l'aeroporto di nome *nome*.

- **nuova_compagnia**(*nomefile*)

Legge dal file *nomefile* il nome di una compagnia che sarà aggiunta a \mathcal{C} , con l'elenco di tutte le tratte servite: per ogni tratta il file specifica la sua lunghezza, il suo costo e il numero di jet-points che conferisce. Se esiste già una compagnia con lo stesso nome specificato nel file, cancella tutti i dati relativi a questa compagnia e li rimpiazza con i dati specificati nel file. Se il file *nomefile* non esiste, stampa un messaggio opportuno. È compito dello studente specificare completamente il formato del file.

- **nuova_tratta**(*nome_c*, *nome_a*, *nome_b*, *l*, *c*, *j*)

Aggiunge alle tratte offerte dalla compagnia *nome_c* la tratta dall'aeroporto *nome_a* all'aeroporto *nome_b*. Tale tratta è lunga *l* km, costa *c* \$ e conferisce *j* jet-points. Se non esiste alcuna compagnia di nome *nome_c*, la crea. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, li crea. Se già esiste una tratta fra *nome_a* e *nome_b* offerta da *nome_c*, ne aggiorna lunghezza, costo e jet-points.

- **viaggio_economico**(*nome_a*, *nome_b*)

Stampa un viaggio di costo minimo fra gli aeroporti *nome_a* e *nome_b*. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.

- **viaggio_corto**(*nome_a*, *nome_b*)

Stampa un viaggio di lunghezza minima fra gli aeroporti *nome_a* e *nome_b*. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.

- **viaggio_diretto**(*nome_a*, *nome_b*)

Stampa un viaggio con numero minimo di scali fra gli aeroporti *nome_a* e *nome_b*. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.

- **piano_voli**(*nome*)
 Stampa un piano di voli per la compagnia *nome*. Se non esiste alcun siffatto piano di voli, stampa un messaggio opportuno. Se la compagnia *nome* non esiste, stampa un messaggio opportuno. È compito dello studente specificare il formato dell'output.
- **elimina_tratta**(*nome_c, nome_a, nome_b*)
 Cancella la tratta da *nome_a* a *nome_b* offerta dalla compagnia *nome_c*, se siffatta tratta esiste.
- **fallimento_compagnia**(*nome*)
 Cancella la compagnia *nome* e tutti i dati ad essa relativi, se tale compagnia esiste.
- **chiusura_aeroporto**(*nome*)
 Cancella l'aeroporto *nome* e tutti i dati ad esso relativi, se tale aeroporto esiste.
- **nuovo_partenariato**(*nomefile*)
 Legge dal file *nomefile* l'insieme dei nomi delle compagnie che formano un nuovo partenariato. Se qualche compagnia non esiste, la crea. Se una compagnia appartiene già a un partenariato, la rimuove da quest'ultimo e la aggiunge al partenariato specificato dal file. Se il file *nomefile* non esiste, stampa un messaggio opportuno. È compito dello studente specificare completamente il formato del file.
- **fusione_partenariati**(*nome_c, nome_d*)
 Crea un nuovo partenariato formato da tutti i partner di *nome_c* e tutti i partner di *nome_d*. Cancella gli eventuali partenariati a cui appartenevano *nome_c* e *nome_d*.
- **stampa_partenariato**(*nome*)
 Stampa tutte le compagnie che appartengono allo stesso partenariato della compagnia *nome*.
- **viaggio_economico_partenariato**(*nome_a, nome_b*)
 Stampa un viaggio di costo minimo fra gli aeroporti *nome_a* e *nome_b*, le cui tratte appartengano tutte a compagnie di uno stesso partenariato. Stampa il numero di jet-points acquisiti. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.
- **viaggio_corto_partenariato**(*nome_a, nome_b*)
 Stampa un viaggio di lunghezza minima fra gli aeroporti *nome_a* e *nome_b*, le cui tratte appartengano tutte a compagnie di uno stesso partenariato. Stampa il numero di jet-points acquisiti. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.
- **viaggio_jetpoints**(*nome_a, nome_b*)
 Stampa un viaggio che massimizzi i jet-points acquisiti fra gli aeroporti *nome_a* e *nome_b*, le cui tratte appartengano tutte a compagnie di uno stesso partenariato. Stampa il numero di jet-points acquisiti. Se non esiste un siffatto viaggio, stampa un messaggio opportuno. Se uno o entrambi gli aeroporti di nome *nome_a* e *nome_b*, non esistono, stampa un messaggio opportuno.
- **acquista_swindle**(*j, nomefile*)
 Stampa il massimo valore in swindles acquistabili con *j* jet-points, secondo la tabella *swindle/jet-points* specificata nel file *nomefile*. Se il file *nomefile* non esiste, stampa un messaggio opportuno. È compito dello studente specificare completamente il formato del file.

3 Modalità di consegna

La presente traccia è valida per l'appello dell'11 febbraio 2013.

La relazione (non meno di 3, non più di 10 pagine in formato pdf o rtf) va inviata per posta elettronica all'indirizzo `aguzzoli@dsi.unimi.it` entro il 18 febbraio 2013.

La relazione e gli altri file aggiuntivi (file sorgenti C, esempi di input, ecc) devono essere contenuti in un unico archivio `.zip` il cui nome dovrà essere della forma `cognome_matricola.zip`. Tutti i file nell'archivio, compresa la relazione, devono riportare nome, cognome e matricola dell'autore.

In generale non è prevista una discussione orale dei progetti, ma in casi particolari potranno essere richiesti dei chiarimenti via mail o dal vivo all'autore delle relazione.

La realizzazione del progetto è una prova d'esame da svolgersi **individualmente**. I progetti giudicati frutto di **copiatura** saranno **estromessi** d'ufficio dalla valutazione.

La versione aggiornata del progetto è pubblicata in `.pdf` sul sito:

<http://homes.di.unimi.it/~aguzzoli/algo.html>.

Si consiglia di consultare periodicamente questo sito per eventuali correzioni e/o precisazioni relative al testo del progetto.