

Laboratorio di Algoritmi

Progetto “Formazione” (luglio 2024)

Nota: La scadenza del progetto è fissata per martedì 9 luglio **compreso**.

Nota: Si consiglia di consultare sulla pagina web il documento che riporta le avvertenze utili per lo svolgimento del progetto. Si consiglia anche di verificare di tanto in tanto gli aggiornamenti a questo documento, che potranno riportare risposte ai dubbi degli studenti e correzioni di eventuali errori.

Il problema Un’azienda vuole definire in maniera algoritmica, anziché a mano, il piano operativo dei propri progetti a medio termine. Infatti, l’azienda ha un numero abbastanza elevato di progetti e di dipendenti il piano deve tener presenti e aspetti molteplici che interagiscono in modi complicati. Per esempio, ogni progetto ha una data di avvio e una data di scadenza imposte dall’esterno. Dal progetto ci si stima di ottenere un certo guadagno, in base al quale si vuole decidere quali progetti cancellare se non fosse possibile portarli tutti a termine. Ogni progetto richiede un insieme di competenze diverse, ciascuna a un dato livello. Corrispondentemente, gli impiegati possiedono competenze disparate a livelli ineguali. Quindi, gli impiegati che prendono parte a un progetto non sono intercambiabili, ma devono essere assegnati a un ruolo preciso.

Più formalmente, indicheremo con P l’insieme di progetti che l’azienda intende affrontare. Ciascun progetto $p \in P$ è caratterizzato da una durata d_p espressa in giorni interi, un giorno di avvio a_p (prima del quale non si può lavorare) e uno di scadenza s_p (entro il quale il progetto deve terminare) e un guadagno atteso g_p . I giorni e il guadagno sono espressi da numeri interi non negativi. Il progetto p richiede una combinazione di competenze C_p , tratte da un insieme generale C di competenze. Ogni competenza della combinazione ha un livello minimo r_{pc} richiesto all’impiegato che coprirà il ruolo corrispondente, espresso come numero intero positivo¹. La combinazione C_p può includere più volte la stessa competenza, con livelli diversi o anche uguali. Per esempio, un progetto può richiedere tre impiegati con “conoscenza del linguaggio C”, uno a livello ≥ 5 e due a livello ≥ 2 . Gli impiegati formano un insieme I , e ciascuno possiede un sottoinsieme $C^{(i)} \subseteq C$ di competenze (ovviamente, senza ripetizioni), ciascuna con un livello l_{ic} ².

Tutto questo richiede alcune analisi gestionali per descrivere la domanda e l’offerta disponibile di competenze. La prima analisi gestionale richiede di valutare quanto siano stringenti i vincoli temporali sui progetti. Ogni progetto va svolto senza interruzioni, partendo all’inizio di un giorno che non preceda strettamente il giorno di avvio e terminando alla fine di un giorno che non segua strettamente il giorno di scadenza. La durata effettiva e le date di avvio e scadenza sono sempre tali da consentire l’esecuzione del progetto, ma possono lasciare un margine, che consenta di ritardare il giorno di partenza oppure di anticipare il giorno di termine. Si definiscono più stringenti i progetti con margine inferiore (in caso di parità, quelli di durata superiore).

La seconda analisi gestionale richiede di valutare le competenze più scarse in relazione ai progetti attualmente disponibili. A tale fine, si possono grossolanamente considerare i livelli di competenza posseduti dai vari impiegati come punteggi, la cui somma fornisce una misura complessiva di offerta, e i livelli di competenza richiesti

¹Il livello massimo non è definito a priori: se influisce sulla complessità temporale o spaziale dell’algoritmo, va considerato come uno dei parametri del problema.

²Per non complicare troppo l’analisi, si consideri il numero delle competenze richieste da ogni progetto e quello delle competenze possedute da ogni impiegate come limitati solo dal numero totale di competenze

da ciascun progetto come punteggi, la cui somma fornisce una misura complessiva di domanda. Il rapporto fra le misure della domanda e dell'offerta esprime la scarsità della competenza stessa.

La terza analisi gestionale riguarda la possibilità di istituire un meccanismo di *formazione*, consistente nel determinare coppie di impiegati e assegnare al primo il ruolo di istruttore e al secondo quello di apprendista. L'istruttore deve possedere almeno una competenza comune con l'apprendista (si esclude cioè di fornire da zero competenze nuove) e il suo livello deve superare di almeno 2 punti quello dell'apprendista. Si vuole per prima cosa capire se questo meccanismo consenta di creare reti di *formazione*, determinando i sottoinsiemi di impiegati che sono *reciprocamente* raggiungibili attraverso catene di trasferimento, in cui un impiegato insegna una competenza a un altro, questo una competenza (anche diversa) a un terzo, ecc. . . . , fino a raggiungere ogni altro membro del sottoinsieme, in ambo i versi.

Ciò fatto, si procederà a valutare un possibile meccanismo di *formazione*, con il quale gli impiegati possano trasmettersi competenze a vicenda. Questo passo consiste nello scegliere fra le coppie potenziali di impiegati (istruttore, apprendista) un sottoinsieme ottimale di coppie distinte tra loro, in modo che nessun impiegato appartenga a più di una coppia, in uno o nell'altro ruolo. L'ottimalità viene definita associando ad ogni coppia potenziale un peso, pari alla somma su tutte le competenze passibili di trasferimento della differenza fra il livello di tale competenza posseduto dal istruttore e quello posseduto dall'apprendista. Esiste un algoritmo polinomiale per determinare il sottoinsieme di coppie disgiunte di peso totale massimo, ma tale algoritmo eccede i limiti del corso. Quindi, lo sostituiremo con un'euristica *greedy*, che aggiunge via via alla soluzione la coppia di impiegati di peso massimo (in caso di parità, prevale la coppia con il primo istruttore in ordine alfabetico³; se l'istruttore è lo stesso, quella col primo apprendista).

Per concludere, si passerà dal lungo periodo al piano operativo, determinando un assegnamento degli impiegati ai ruoli richiesti dai vari progetti e un calendario per i progetti stessi, in modo da scegliere quali svolgere e quali cancellare. Quest'ultima fase considera solo le competenze attuali degli impiegati, non quelle che si potranno acquisire dopo la formazione. La determinazione del piano deve rispettare le seguenti regole. Ogni impiegato può essere assegnato a un solo progetto per volta, seguendolo dal principio alla fine, in un ruolo che impegni esattamente una delle sue competenze e che richieda un livello di tale competenza non superiore a quello che l'impiegato possiede. Lo scopo è portare a termine l'insieme di progetti che porti il massimo guadagno all'azienda. Si tratta di un problema per il quale non sono noti algoritmi esatti polinomiali. Si procederà quindi col seguente algoritmo euristico. Si considereranno i progetti per giorno di avvio crescente (in caso di parità, per guadagno decrescente; in caso di ulteriore parità, in ordine alfabetico). Per ogni progetto si considereranno le competenze richieste in ordine di livello non crescente (in caso di parità, ordine alfabetico⁴). Per ciascuna si cercherà l'impiegato disponibile (non già assegnato ad altri progetti nei giorni impegnati dal progetto corrente) che possieda la competenza stessa a un livello non inferiore a quello richiesto e che sia il più "spendibile". Con questo si intende che il livello con cui possiede la competenza corrente eccede il meno possibile il livello richiesto; in caso di parità, si sceglierà l'impiegato con la minima somma di tutti i livelli di competenza posseduti, e in caso di ulteriore parità il primo in ordine alfabetico. Se

³Per semplicità, siccome i nomi dei progetti, delle competenze e di alcuni impiegati contengono caratteri non alfabetici e le lettere sono sia minuscole sia maiuscole, intendiamo sempre come "ordine alfabetico" quello dei caratteri nella tabella ASCII.

⁴Una competenza può essere richiesta più volte con lo stesso livello, ma in questo caso l'ordine è ininfluenza.

almeno una competenza non è disponibile, il progetto viene rifiutato; altrimenti, gli impiegati selezionati vengono assegnati al progetto, che li impegna dal giorno di avvio a quello di termine (valutato in base alla durata) compresi.

Il progetto Il progetto richiede la stesura di un programma che legga le informazioni necessarie da tre file di testo⁵. Il primo file fornisce l'elenco delle competenze possibili. Si apre con il loro numero e prosegue con una riga per ciascuna competenza. Per semplicità, ogni competenza è descritta da una sola parola, di al più 50 caratteri⁶. Per esempio:

```
7
XML
TechWriting
HTML
C++
Algorithms
Programming
Forecasting
```

indica che sono state individuate 7 diverse competenze, di cui la prima è `XML`, la seconda `TechWriting`, e così via.

Il secondo file fornisce l'elenco dei progetti nei quali l'azienda è impegnata. La prima riga del file ne fornisce il numero. Segue un blocco di righe per ciascun progetto:

- la prima riga del blocco riporta:
 - il nome del progetto (una singola parola di al più 50 caratteri);
 - la durata in giorni d_p (numero intero positivo);
 - il giorno di avvio a_p (indice intero positivo);
 - il giorno di scadenza s_p (indice intero positivo);
 - il guadagno g_p (numero intero positivo);
 - il numero di ruoli, cioè singole competenze richieste;
- ciascuna riga seguente si riferisce a una richiesta e riporta:
 - il nome della competenza richiesta;
 - il livello minimo richiesto r_{pc} (numero intero positivo).

Per esempio:

```
3
WebServer 5 3 10 5 2
HTML 3
C++ 2
Risorse_umane 8 8 15 8 2
Algorithms 2
TechWriting 1
Futuro 7 6 15 10 1
Forecasting 1
```

⁵Violazioni all'ordine dei tre file saranno penalizzate nella valutazione.

⁶Questa indicazione di massima serve a dimensionare le stringhe, non va intesa come un invito a considerare $O(1)$ il numero di possibili competenze nelle analisi di complessità e nemmeno il tempo e lo spazio richiesti a gestirle. Lo stesso vale nel seguito per i progetti e gli impiegati. Per semplicità, però, assumeremo un parametro solo per le tre dimensioni.

indica che 3 progetti sono disponibili. Il progetto `WebServer` dura 5 giorni, va avviato non prima del giorno 3 e completato entro il giorno 10, fornisce un guadagno pari a 5 e richiede 2 impiegati: il primo deve avere competenze di `HTML` a livello ≥ 3 , il secondo competenze di `C++` a livello ≥ 2 . Il progetto `Risorse_umane` dura 8 giorni, va avviato non prima del giorno 8 e completato entro il giorno 15, fornisce un guadagno pari a 8 e richiede 2 impiegati, che abbiano (rispettivamente) competenze di `Algorithms` a livello ≥ 2 e di `TechWriting` a livello ≥ 1 . Infine, il progetto `Futuro` dura 7 giorni, va avviato non prima del giorno 6 e completato entro il giorno 15, fornisce un guadagno pari a 10 e richiede un impiegato con la competenza in `Forecasting` a livello ≥ 1 .

Il terzo file di testo fornisce i dati relativi alle competenze degli impiegati disponibili. La prima riga indica il loro numero. Segue un blocco di righe per ciascun impiegato:

- la prima riga del blocco riporta:
 - il nome dell'impiegato (una singola parola di al più 50 caratteri);
 - il numero di competenze possedute;
- ciascuna riga seguente si riferisce a una competenza e riporta:
 - il nome della competenza richiesta;
 - il livello posseduto l_{ic} (numero intero positivo).

Per esempio:

```
3
Anna 2
C++ 4
TechWriting 1
Bob 2
HTML 4
Programming 1
Christine 3
Algorithms 4
Programming 5
HTML 1
```

indica che sono disponibili 3 impiegati: `Anna` ha 2 competenze, cioè `C++` a livello 4 e `TechWriting` a livello 15; `Bob` ha competenza in `HTML` a livello 4 e `Programming` a livello 1; `Christine` ha competenza in `Algorithms` a livello 4, in `Programming` a livello 5 e in `HTML` a livello 1.

Il progetto richiede di valutare la stringenza di ogni progetto, definita come il numero di giorni di margine con cui si può ritardare la partenza del progetto o anticipare il termine rispettando i giorni di avvio e scadenza e la durata indicati nei dati. Il programma deve stampare una prima riga con il numero totale di progetti, seguito dall'elenco dei progetti stessi, ordinati per margine non decrescente. A parità di margine, si ordina per durata non crescente; a parità di entrambi, si usa l'ordine alfabetico. Di ogni progetto si stamperà il nome, seguito dal margine e dalla durata. Nell'esempio:

```
3
Risorse_umane 0 8
Futuro 3 7
WebServer 3 5
```

indica che ci sono 3 progetti: il più stringente è `Risorse_umane`, che ha margine nullo e durata pari a 8 giorni; il secondo è `Futuro`, che ha 3 giorni di margine e 7 di durata; il meno stringente è `WebServer`, che ha anch'esso 3 giorni di margine, ma solo 5 giorni di durata.

La valutazione della scarsità delle competenze comporta di calcolare le misure di offerta e di domanda sopra descritte (somma dei livelli con cui ogni competenza è posseduta da tutti gli impiegati e richiesta da tutti i progetti) e farne il rapporto. Il programma stamperà una prima riga con il numero di competenze richieste (in genere, inferiore al numero totale di competenze). Quindi, stamperà riga per riga tali competenze ordinate per valore non crescente del rapporto fra domanda e offerta (usando, in caso di parità, l'ordine alfabetico). Di ogni competenza stamperà il nome, il valore del rapporto arrotondato a 3 cifre decimali, il numero di richieste per tale competenza (eventualmente, più richieste per lo stesso progetto) e il numero di impiegati che la possiedono. Si noti che il rapporto è indefinito quando la misura di offerta è nulla. Queste competenze, se presenti, vanno stampate per prime, in ordine alfabetico, sostituendo il rapporto con il simbolo ND (che sta per “non disponibile”). Nell'esempio:

```
5
Forecasting ND 1 0
TechWriting 1.000 1 1
HTML 0.600 1 2
Algorithms 0.500 1 1
C++ 0.500 1 1
```

indica che solo 5 delle 7 competenze sono richieste (mancano `Programming` e `XML`), che `Forecasting` non è disponibile, in quanto è richiesta da un progetto, ma nessun impiegato la possiede. `TechWriting` ha rapporto fra domanda e offerta pari a 1.000, essendo richiesta da un progetto con livello 1 e posseduta da un impiegato con livello 1. `HTML` ha rapporto pari a 0.600, dato che viene richiesta da un progetto con livello 3 ed è posseduta da un impiegato con livello 4 e uno con livello 1. Infine, `Algorithms` e `C++` hanno rapporto pari a 0.500 e sono entrambe richieste da un progetto e possedute da un impiegato (in entrambi i casi con livello 2 richiesto e 4 posseduto); vengono quindi stampate in ordine alfabetico.

L'analisi della rete di *formazione* potenziale comporta di determinare i sottoinsiemi di impiegati reciprocamente connessi dalla possibilità di trasmettersi a vicenda, direttamente o indirettamente, competenze. Il programma stamperà in una prima riga il numero di sottoinsiemi seguito dalla parola chiave **gruppi**⁷. Ogni riga seguente riguarderà un gruppo e conterrà la sua cardinalità e l'elenco degli impiegati in ordine alfabetico. I gruppi saranno ordinati per ordine alfabetico del primo impiegato. Nell'esempio:

```
2 gruppi
1 Anna
2 Bob Christine
```

dato che `Anna` è isolata, mentre `Bob` può insegnare `HTML` a `Christine` e questa può insegnare `Algorithms` a `Bob`. Si noti che la struttura rimarrebbe la stessa se `Anna` potesse insegnare una competenza a uno degli altri impiegati, oppure impararne una da loro, mentre gli impiegati formerebbero un gruppo unico se `Anna` potesse sia insegnare una competenza sia impararne una. Questo varrebbe sia se lo scambio avvenisse con entrambi, sia se avvenisse con un impiegato in un verso e con l'altro nel verso opposto, e infine anche se avvenisse in entrambi i versi con un solo impiegato,

⁷Per semplicità, sempre al plurale, anche se vi fosse un solo gruppo.

perché tutti questi casi creerebbero una catena di trasmissione fra ogni coppia di impiegati.

La fase effettiva di *formazione* comporta di scegliere fra tutte le coppie potenziali (istruttore, apprendista) un sottoinsieme di coppie disgiunte di peso totale massimo, ottenuto con un algoritmo *greedy*. Il programma deve stampare una prima riga contenente il numero di coppie seguito dalla parola chiave **coppie**⁸. A questa riga segue una riga per ogni gruppo, che riporti i nomi dell'istruttore e dell'apprendista, in ordine alfabetico rispetto all'istruttore, il peso della coppia e il numero di competenze che potrebbero trasmettersi (nel verso indicato dalla coppia). Nell'esempio:

```
1 coppie
Christine Bob 4 1
```

dato che la coppia (Bob,Christine) ha peso $4 - 1 = 3$ e la coppia (Christine,Bob) ha peso $5 - 1 = 4$, per cui prevale la seconda, dopo di che la prima non è più accettabile.

Infine, l'assegnamento dei progetti in base all'algoritmo euristico sopra descritto verrà stampato a video nel formato seguente. Per prima cosa, si stamperà il numero di progetti eseguibili seguito dalle parole chiave **progetti su** e dal numero totale di progetti disponibili⁹. Nella riga seguente si stamperà la parola chiave **guadagno** seguita dal guadagno totale atteso dai progetti previsti. Quindi, si stamperanno i progetti in ordine alfabetico, ciascuno in una riga contenente il nome del progetto seguito dall'elenco alfabetico degli impiegati assegnati ad esso. Per i progetti non svolti, si stamperà la parola chiave **rifiutato** anziché l'elenco degli impiegati. Nell'esempio:

```
2 progetti su 3
guadagno 13
Futuro rifiutato
Risorse_umane 8 15 Anna Christine
WebServer 3 7 Anna Bob
```

indica che 2 progetti su 3 vengono svolti, con un guadagno atteso pari a 13 (8 per **Risorse_umane** e 5 per **WebServer**). Il progetto **Futuro** viene rifiutato, mentre **Risorse_umane** impegna **Anna** e **Christine** dal giorno 8 al giorno 15 inclusi e **WebServer** impegna **Anna** e **Bob** dal giorno 3 al giorno 7 inclusi.

Chiarimenti

In questa sezione saranno riportate le risposte a domande e dubbi.

⁸Per semplicità, sempre al plurale, anche nel caso di una sola coppia o nessuna.

⁹Anche in questo caso, per semplicità si userà il plurale anche se si svolge un solo progetto o nessuno.