Laboratorio di Algoritmi

Progetto "Impiegati" (novembre 2024)

Nota: La scadenza del progetto è fissata per domenica 3 novembre **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 La gestione delle risorse umane nelle grandi aziende è un problema complicato e multiforme, che combina aspetti temporali, quantitativi e qualitativi. I progetti, infatti, si svolgono nel tempo, con durate e scadenze (più o meno rigide) diverse tra loro. Anche l'importanza dei progetti in generale non è la stessa. Ovviamente, un'azienda punta a portare a termine tutti i progetti su cui è impegnata, possibilmente nel rispetto delle relative scadenze. Nel caso in cui ciò non fosse possibile, però, andrà fatta una scelta riguardo i progetti da ritardare o, al limite, cancellare.

Dal punto di vista delle risorse umane, gli impiegati vengono assegnati ai progetti in modo da coprire non solo la quantità di lavoro richiesta, ma anche il tipo di competenze necessarie a svolgerlo, non essendo in generale intercambiabili. Anche considerando un solo tipo di competenza, il livello con cui il singolo impiegato la possiede può essere sufficiente per progetti più elementari, ma non per progetti più avanzati. Infine, la disponibilità di un impiegato di prestare la propria opera in un progetto è limitata in ogni singolo giorno, e dipende quindi dal calendario fissato per i progetti ai quali viene assegnato. È anche noto che suddividere eccessivamente la propria opera su progetti diversi riduce la concentrazione e peggiora la qualità e quantità del lavoro.

Avremo quindi un insieme di progetti P, un insieme di impiegati I e un insieme di competenze C. Ciascun progetto $p \in P$ sarà caratterizzato da una durata d_p espressa in giorni, un giorno di scadenza s_p (entro il quale il progetto deve terminare) e un guadagno g_p. 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 positivo[1]. La combinazione 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]. È possibile che un progetto richieda competenze non disponibili.

Dal punto di vista gestionale, occorre un'analisi preliminare dei dati, che ne estragga informazioni sul carico di lavoro richiesto dai progetti e sulle potenzialità offerte dagli impiegati. Bisognerà per prima cosa costruire un quadro generale della relazione fra domande e offerta di competenze, indicando per per ciascun livello minimo di ciascuna competenza quanti impiegati dispongono e quanti progetti richiedono un livello almeno pari ad esso. Questo perché, naturalmente, un impiegato possiede una competenza a un dato livello se la possiede ad un livello non inferiore e un progetto richiede una competenza a un dato livello se la richiede ad un livello

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

[2] Per 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

non inferiore. Quindi, si dovrà tratteggiare la distribuzione temporale dei progetti per prevedere i periodi di maggiore e minore carico. Ipotizzando per semplicità una strategia "just-in-time" di calendarizzazione dei progetti, si supporrà di avviare ogni progetto nell'ultimo giorno utile a concluderlo il giorno della scadenza. Questo determinerà un calendario che consente di indicare giorno per giorno il numero di progetti attivi e le date di inizio e termine di ciascun progetto. Quindi, si analizzerà la relazione fra impiegati e progetti, indicando per ciascun impiegato l'elenco dei progetti ai quali ha competenza sufficiente per prender parte. Infine, le relazioni fra impiegati indotte indirettamente dai progetti a cui possono attivamente partecipare e dai livelli di competenza di cui dispongono permetteranno di eseguire due analisi finali. La prima è volta a individuare impiegati che potrebbero formare gruppi di lavoro: due impiegati sono giudicati "compatibili" quando hanno almeno un progetto in comune su cui possono lavorare. La seconda ha lo scopo di determinare se le competenze disponibili tendono ad essere raggruppate in pochi individui o distribuite, e se gli individui più esperti tendono a presentare insiemi di competenze simili fra loro o complementari. Questo consentirà di individuare meglio i punti di debolezza su cui poter intervenire in futuro con un'opportuna formazione o con assunzioni mirate. L'analisi assegna ad ogni impiegato un peso determinato dalla somma dei suoi livelli di competenza su ogni materia e stabilisce una sovrapposizione fra due impiegati quando hanno almeno una competenza in comune (con qualsiasi livello positivo). Trovare il sottoinsieme di impiegati di peso massimo che non abbiano competenze in comune permette di capire se i nuclei in cui si concentrano le competenze dell'azienda sono molti o pochi e come sono costituiti. Siccome il problema non ammette algoritmi polinomiali, verrà risolto con un algoritmo *greedy*. L'algoritmo banale partirebbe dall'impiegato con peso massimo e aggiungerebbe via via gli impiegati di peso massimo privi di competenze in comune con tutti i precedenti. Questo ignora però il fatto che un impiegato con molte competenze comuni ad altri eliminerebbe rapidamente molti elementi potenziali della soluzione, mentre impiegati con peso inferiore, ma competenze non sovrapposte, si presentano come candidati migliori. L'algoritmo *greedy* userà quindi come criterio di scelta fondamentale il rapporto fra il peso di un impiegato e il numero di altri impiegati con competenze sovrapposte al primo (incrementato di 1 per evitare la divisione per 0 che si avrebbe nel caso in cui non ci fossero altri impiegati). In caso di parità, si sceglierà l'impiegato di peso massimo, e in caso di ulteriore parità il primo in ordine alfabetico.

Il progetto Il progetto richiede la stesura di un programma che legga le informazioni necessarie da tre file di testo. 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 caratteri[3]. Per esempio:

```
6
XML
TechWriting
HTML
C++
Algorithms
Programming
```

[3]Questa indicazione di massima serve a dimensionare le stringhe, non va intesa come un invito a banalizzare le analisi di complessità ignorando i tempi di lettura e scrittura delle stringhe, né tanto meno considerando costante il numero di competenze possibili. Lo stesso vale nel seguito per i progetti e gli impiegati.

indica che sono state individuate 6 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 di ogni 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 scadenza s_p (indice intero positivo);
  - il guadagno g_p (numero intero positivo);
  - il numero di impiegati, dunque di singole specifiche competenze e livelli richiesti;
- ciascuna riga seguente si riferisce a un impiegato richiesto e riporta
  - il nome della competenza richiesta;
  - il livello minimo richiesto, r_{pc} (numero intero positivo).

Per esempio:

```
2
WebServer 5 10 5 2
HTML 3
C++ 2
Risorse_umane 8 15 8 3
Algorithms 2
Programming 1
TechWriting 1
```

indica che 2 progetti sono attivi. Il progetto `WebServer` dura 5 giorni, va 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 `Progetti` dura 8 giorni, va completato entro il giorno 15, fornisce un guadagno pari a 8 e richiede 3 impiegati, che abbiano (rispettivamente) competenze di `Algorithms` a livello ≥ 2, di `Programming` e di `TechWriting` 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 di ogni 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 1
HTML 3
Christine 1
Algorithms 2
```

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

Come già descritto, il progetto richiede di valutare per ogni competenza richiesta o disponibile e per ogni possibile livello (da 1 al massimo richiesto o disponibile) quanti impiegati sono richiesti dai progetti e quanti sono potenzialmente disponibili[4]. Queste informazioni vanno riportate in ordine alfabetico rispetto alle competenze[5] e per valori crescenti rispetto al livello. Ogni competenza formerà un blocco di righe che comincia con il nome della competenza e prosegue con una riga per ciascun livello di interesse, descritto dalla parola chiave `livello`, seguita da uno spazio, l'indice del livello, due punti, uno spazio, la disponibilità complessiva offerta dagli impiegati preceduta da `o`, uno spazio, un trattino, un altro spazio, e infine la richiesta complessiva posta dai progetti preceduta da `d`. Nell'esempio:

```
Algorithms
livello 1:  o1 - d1
livello 2:  o1 - d1
C++
livello 1:  o1 - d1
livello 2:  o1 - d1
livello 3:  o1 - d0
livello 4:  o1 - d0
HTML
livello 1:  o1 - d1
livello 2:  o1 - d1
livello 3:  o1 - d1
Programming
livello 1:  o0 - d1
TechWriting
livello 1:  o1 - d1
```

indica che la competenza `Algorithms` a livello ≥ 1 è offerta da 1 impiegato e i progetti richiedono 1 impiegato che possieda. Lo stesso avviene per livelli ≥ 2 della stessa competenza e per i livelli ≥ 1 e ≥ 2 della competenza `C++`. Per i livelli ≥ 3 e ≥ 4, invece, esiste un impiegato che ne dispone, ma la richiesta attuale è nulla.

Il calendario di massima dei progetti si aprirà con una riga contenente la parola chiave `Calendario`, seguita da una riga per ogni giorno, contenente l'indice del giorno e il numero di progetti attivi, eventualmente seguito prima dai nomi dei progetti avviati, poi da quelli dei progetti terminati in quel giorno. Ciascun nome va preceduto da `<` se il progetto parte e seguito da `>` se si conclude, e i due elenchi di nomi vanno riportati in ordine alfabetico. Il calendario parte dall'avvio del primo progetto e si conclude con la scadenza dell'ultimo. Per esempio:

```
Calendario
6 1 <WebServer
7 1
```

[4] "Potenzialmente" perché potrebbero esistere, ma essere impegnati in altri progetti.

[5] Per semplicità, intendiamo sempre come "ordine alfabetico" quello dei caratteri nella tabella ASCII.

```
8 2 <Risorse_umane
9 2
10 2 WebServer>
11 1
12 1
13 1
14 1
15 1 Risorse_umane>
```

indica che nel giorno 6 parte il primo progetto (`Webserver`), che è attivo anche nel giorno seguente. Il giorno 8 i progetti attivi diventano due, perché parte anche `Risorse_umane`. I progetti attivi rimangono 2 fino al giorno 10, quando si conclude `Webserver`, e poi 1 fino al giorno 15, quando termina `Risorse_umane` e il calendario si chiude.

La relazione fra impiegati e progetti deve essere riportata stampando due righe per ogni impiegato (in ordine alfabetico): la prima riga conterrà il nome, la seconda il numero di progetti potenzialmente compatibili e il loro elenco in ordine alfabetico. Nel consueto esempio:

```
Anna
2 WebServer Risorse_umane
Bob
1 WebServer
Christine
1 Risorse_umane
```

indica che `Anna` può lavorare su entrambi i progetti, `Bob` solo sul progetto `Webserver` e `Christine` solo sul progetto `Risorse_umane`.

Deve poi seguire l'elenco delle compatibilità fra impiegati, introdotto da una riga con la sola parola chiave `Compatibilita'`, a cui seguono, riga per riga, le coppie di impiegati compatibili (cioè con almeno un progetto su cui potrebbero entrambi lavorare[6]). Le coppie di impiegati vanno indicate in ordine lessicografico, cioè ordinando i due nomi di ogni coppia alfabeticamente, e le coppie in base all'ordine alfabetico sul primo e poi sul secondo nome. Nel nostro esempio:

```
Compatibilita'
(Anna,Bob)
(Anna,Christine)
```

indica che `Anna` può lavorare con `Bob` e con `Christine`, ma questi ultimi non possono lavorare allo stesso progetto.

Infine, l'algoritmo *greedy* per determinare euristicamente il sottoinsieme di impiegati di peso massimo con competenze totalmente disgiunte ha una soluzione banale nell'esempio considerato, perché tutti gli impiegati hanno competenze diverse. Quindi, l'algoritmo sceglie prima `Anna` che ha peso $4 + 1 = 5$ e un numero di sovrapposizioni pari a 0, da cui un rapporto pari a $5/(0 + 1) = 5$, mentre `Bob` ha rapporto pari a $3/1 = 3$ (peso 3 e nessuna sovrapposizione) e `Christine` ha rapporto pari a $2/1 = 2$ (peso 2 e nessuna sovrapposizione). Al passo successivo, dopo aver cancellato `Anna` e tutti i candidati con sovrapposizioni (in questo caso, nessuno), l'algoritmo sceglie `Bob`. Al terzo passo, sceglie `Christine`. Il peso totale della soluzione è quindi $5 + 3 + 2 = 10$. La soluzione deve essere stampata introducendola con una riga che riporta la parola chiave `Soluzione` e il suo peso totale, seguita riga per riga dai nomi degli impiegati scelti e dai loro pesi, nell'ordine con cui sono stati scelti. Quindi:

[6] Per semplicità, ignoriamo il fatto che i due impiegati potrebbero essere compatibili col progetto, ma solo se assegnati allo stesso ruolo, e quindi essere in pratica incompatibili tra loro.

```
Soluzione 10
Anna 5
Bob 3
Christine 2
```

Per maggiore chiarezza, consideriamo un altro esempio, con insieme di impiegati $I = \{i_1, i_2, i_3, i_4\}$, rispettivamente di peso 4, 2, 3 e 1, e supponiamo che i_1 e i_2 abbiano una o più competenze sovrapposte, e lo stesso avvenga per i_1 e i_3. I rapporti risultanti sono 4/3 per i_1 (peso 4 e 2 impiegati con competenze sovrapposte), 2/2 per i_2, 3/2 per i_3 e 1/1 per i_4. Al primo passo, l'algoritmo sceglie i_3. Al secondo, sceglie i_2, perché i_1 ha competenze sovrapposte con i_3 e i_4 ha lo stesso rapporto, ma peso inferiore. Al terzo passo, sceglie i_4 e termina.

Chiarimenti

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