

Laboratorio di Algoritmi

Progetto “Elezioni” (giugno 2025)

Nota: La scadenza del progetto è fissata per lunedì 9 giugno **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 Gli abitanti di un comune devono decidere in qual modo investire il bilancio comunale nei prossimi anni. Hanno a disposizione un elenco di progetti che intendono ordinare in modo che i fondi disponibili vengano via via assegnati al primo progetto che non sia ancora stato avviato. Vogliono definire l'ordine in modo razionale e democratico al tempo stesso, e il numero di abitanti e di progetti è eccessivo per poter gestire manualmente la procedura. Inoltre, indagando la letteratura sull'argomento, hanno scoperto che vi sono molti metodi per farlo e, cosa che li getta nella più profonda costernazione, i vari metodi producono risultati diversi e nessuno garantisce i requisiti minimi che ci si aspetterebbe da un approccio fondato sulla matematica. Decidono comunque di realizzare algoritmi che applichino alcuni dei metodi più comunemente raccomandati, nella speranza che i risultati concordino almeno parzialmente.

Cominciano quindi raccogliendo le preferenze di tutti gli abitanti. È dato un insieme I di n individui, determinati dal loro codice fiscale, e un insieme P di p progetti, determinati univocamente dal loro nome. La preferenza di ogni individuo $i \in I$ è descritta da un *ordine debole* sull'insieme dei progetti, cioè da una relazione binaria su P riflessiva, completa e transitiva, ma non necessariamente antisimmetrica. Scriveremo $q \preceq_i q'$ quando l'individuo i preferisce debolmente q a q' , il che può indicare che i preferisce strettamente il primo ($q \prec_i q'$) o che i due progetti gli sono indifferenti ($q \sim_i q'$). Una preferenza corrisponde a una permutazione dell'insieme dei progetti con l'ulteriore indicazione se ciascuno è strettamente preferito o indifferente rispetto al successivo.

Raccolte le preferenze, il metodo più semplice per porle in classifica è noto come *plurality system*, e consiste nel contare per ciascun progetto $q \in P$ il numero di individui che lo considerano strettamente preferibile o almeno indifferente rispetto a ogni altro progetto: $V(q) = \{i \in I : q \preceq_i q', \forall q' \in P\}$. Questo corrisponde a consentire ad ogni individuo di attribuire un voto a tutti i progetti che gli risultano indifferenti fra loro e strettamente preferiti a tutti gli altri e sommare i voti su tutti gli individui. I progetti vengono quindi ordinati per valori decrescenti del numero di voti ricevuti. Questo determina la preferenza dell'intero gruppo: progetti con più voti sono preferiti strettamente a progetti con meno voti, mentre progetti con lo stesso numero di voti sono indifferenti:

$$q \preceq q' \Leftrightarrow V(q) \geq V(q')$$

Si dimostra facilmente che il risultato è un ordine debole.

Il *metodo di Borda* è un sistema analogo, ma più raffinato. Consiste nell'incaricare ogni individuo i di assegnare ad ogni progetto q un numero di punti pari al numero di progetti rispetto ai quali esso è preferito o indifferente: $B_i(q) = |\{q' \in P : q \preceq_i q'\}|$, sommare per ogni progetto il numero totale di punti ricevuti da tutti gli individui ($B(p) = \sum_{i \in I} B_i(p)$) e ordinare i progetti per valori decrescenti del numero di punti ricevuti:

$$q \preceq q' \Leftrightarrow B(q) \geq B(q')$$

Anche in questo caso, si dimostra facilmente che il risultato è un ordine debole.

Il *metodo di Condorcet* procede in maniera diversa. Per ogni coppia di progetti determina la preferenza del gruppo confrontando il numero di individui che preferiscono il primo al secondo con il numero di individui che preferiscono il secondo al primo. Se il numero di sostenitori del primo progetto prevale strettamente su quello dei sostenitori del secondo, il gruppo preferisce il primo progetto; se invece prevale strettamente il numero di sostenitori del secondo, il gruppo preferisce il secondo progetto. Infine, se i due numeri coincidono, il gruppo è indifferente fra i due progetti.

$$q \preceq q' \Leftrightarrow |\{i \in I : q \preceq q'\}| \geq |\{i \in I : q' \preceq q\}|$$

dove il risultato non cambia se si contano le preferenze individuali deboli o quelle strette. La relazione binaria su P che ne deriva è certamente riflessiva (tutti sono indifferenti fra un progetto e il progetto stesso) e completa (c'è sempre una preferenza stretta in uno dei due versi oppure indifferenza). Viene però a mancare, in generale, la transitività. Quindi, la preferenza di gruppo non è in genere un ordine debole, ma solo un *preordine*.

Si noti però che quando $q \preceq q'$, anche se q' non è preferibile a q , può accadere che esista una catena di preferenze che collegano indirettamente q' a q , passando per altri progetti. In generale, ci saranno sottoinsiemi di progetti che sono fra loro vicendevolmente raggiungibili: in un verso direttamente, nell'altro tramite una catena di preferenze. I sottoinsiemi massimali che godono di questa proprietà, benché non propriamente indifferenti fra loro, si possono considerare approssimativamente tali (se non altro, per disperazione). Progetti che appartengono a sottoinsiemi massimali diversi sono invece strettamente ordinati in un verso e non nel verso opposto, per cui la preferenza fra sottoinsiemi è inequivocabile. La relazione che ne deriva sui sottoinsiemi di progetti non è solo riflessiva e completa, ma anche transitiva e antisimmetrica, e dunque è un *ordine totale*. Determinare tale ordine può essere utile per classificare i progetti.

Per concludere l'analisi della situazione, si possono valutare le somiglianze fra le preferenze dei vari individui. L'idea è raccogliere gli individui in un dato numero k di gruppi i cui membri abbiano preferenze molto simili fra loro e determinare un rappresentante per ciascun gruppo. Lo scopo è di consentire discussioni ristrette più approfondite alle quali sarebbe poco pratico far partecipare tutti gli abitanti del comune. Ne risulta una delle molte varianti del problema di *clustering*, a ciascuna delle quali corrispondono diversi algoritmi risolutivi. Nel caso specifico, modelliamo il problema come la ricerca di un sottoinsieme $x \subseteq I$ composto da k individui *mediani*, scelti in modo da minimizzare la somma su tutti gli individui j della distanza fra j e l'individuo mediano i più vicino a j :

$$f(x) = \sum_{j \in I} \min_{i \in x} d(i, j)$$

Definiamo la distanza fra due individui come la differenza totale fra i punteggi di Borda che i due individui assegnano ai vari progetti:

$$d(i, j) = \sum_{q \in P} |B_i(q) - B_j(q)|$$

Per questo problema non si conoscono algoritmi polinomiali, ma si può ricorrere a un algoritmo euristico di tipo *greedy*. Questo algoritmo parte da un insieme di mediani x vuoto. Ad ogni passo, determina l'individuo i^* non mediano che, aggiunto all'insieme x corrente rende minima la funzione obiettivo:

$$i^* = \arg \min_{i \in I \setminus x} \sum_{j \in I} \min_{k \in x \cup \{i\}} d(j, k)$$

Quando l'insieme x contiene k individui, l'algoritmo termina.

Il progetto Il progetto richiede la stesura di un programma che legga da un file di testo le informazioni relative alle preferenze individuali di tutti gli abitanti del comune. Il file si apre con una riga che contiene il numero di abitanti, il numero di progetti da mettere in classifica e il numero di gruppi da costruire. Per esempio:

```
5 5 2
```

indica che vi sono $n = 5$ abitanti e $p = 5$ progetti e si vogliono ottenere $k = 2$ gruppi di individui.

Ciascuna riga seguente si riferisce a un singolo individuo e ne riporta il codice fiscale e la preferenza. Il codice fiscale è definito per legge come una stringa di 16 caratteri¹. Il codice fiscale è seguito da uno spazio bianco e da un elenco di progetti. Ogni progetto è separato dal successivo da uno spazio bianco, un carattere che descrive il tipo di preferenza (< per la preferenza stretta e = per l'indifferenza) e un altro spazio bianco. Ogni progetto è descritto da una sola parola, di al più 30 caratteri².

```
XPVQCZ02B71P804C Python < Switch = Bay = DejaVu = Celestial
GWDTBD60C03F701L Celestial < DejaVu = Switch < Python < Bay
HNPFSZ57R55P701N DejaVu < Bay = Python = Celestial < Switch
FJVWBR24L53H609K Switch < Python = Celestial < DejaVu < Bay
JQHGVX42C59G003D DejaVu = Bay < Python < Celestial < Switch
```

indica che l'individuo con codice fiscale XPVQCZ02B71P804C preferisce strettamente il progetto Python ai progetti Switch, Bay, DejaVu e Celestial, che considera fra loro indifferenti. Il secondo individuo, che ha codice fiscale GWDTBD60C03F701L preferisce il progetto Celestial ai progetti DejaVu e Switch, che gli sono indifferenti, e preferisce questi al progetto Python e quest'ultimo al progetto Bay.

Il primo metodo considerato (*plurality system*) determina per ogni progetto il numero di individui che lo preferiscono (o almeno lo considerano indifferente) rispetto a tutti gli altri e ordina i progetti per valori non crescenti di preferenze. Il programma dovrà stampare una riga con la parola chiave **Plurality system**, seguita dalla relativa classifica dei progetti. Ogni riga della classifica riporterà il numero di voti seguito da uno spazio bianco e dall'elenco dei progetti che hanno ricevuto tale numero di voti. Questi progetti andranno ordinati alfabeticamente e separati fra loro da spazi bianchi. Per esempio:

```
Plurality system
2 DejaVu
1 Bay Celestial Python Switch
```

indica che nessun progetto ha ricevuto più di due voti, il progetto DejaVu ne ha ricevuti due e gli altri quattro progetti ne hanno ricevuto uno ciascuno. I voti totali superano il numero di individui perché il quinto individuo ha votato due progetti, dato che li considera indifferenti fra loro ed entrambi strettamente preferibili agli altri.

¹Poiché i codici sono stati generati a caso, il carattere finale di controllo è molto probabilmente sbagliato ($\pi = 25/26$).

²Questa indicazione serve a dimensionare le stringhe in modo semplice. Mentre il codice fiscale ha per definizione lunghezza costante, l'indicazione della lunghezza massima per i nomi dei progetti non va intesa come un invito a considerare costante la lunghezza dei nomi stessi nelle analisi di complessità. Tanto meno vanno considerati limitati (e quindi costanti) il numero degli individui e quello dei progetti in base al limitato numero di codici e di nomi univoci generabili.

Il risultato del metodo di Borda verrà stampato in modo analogo. Per prima cosa si stamperà una riga con la parola chiave **Metodo di Borda**, seguita dalla relativa classifica dei progetti. Ogni riga della classifica riporterà il numero di punti seguito da uno spazio bianco e dall'elenco dei progetti che hanno ricevuto tale numero di punti. Questi progetti andranno ordinati alfabeticamente e separati fra loro da spazi bianchi. Per esempio:

```
Metodo di Borda
20 DeJaVu
19 Celestial
18 Python
15 Bay Switch
```

indica che il progetto **DeJaVu** ha ricevuto 20 punti, il progetto **Celestial** 19 punti, il progetto **Python** 18 punti e i progetti **Bay** e **Switch** 15 ciascuno. Infatti, il progetto **DeJaVu** ha ricevuto 4 punti dal primo e dal secondo individuo, 5 punti dal terzo e dal quinto e 2 punti dal quarto. Si noti infatti che per il primo individuo **Switch** è a pari merito con **Bay**, **DeJaVu** e **Celestial**, dunque è almeno indifferente rispetto a 4 progetti (compreso esso stesso).

Il risultato del metodo di Condorcet è introdotto da una riga con la parola chiave **Metodo di Condorcet**. Segue un elenco totalmente ordinato di gruppi di progetti: ogni gruppo è composto di progetti vicendevolmente raggiungibili o con una preferenza diretta o con una catena indiretta di preferenze. I gruppi devono essere stampati nell'ordine, uno per ogni riga, riportando i progetti che ne fanno parte in ordine alfabetico, separati da uno spazio bianco. Per esempio:

```
Metodo di Condorcet
Celestial DeJaVu Python
Bay Switch
```

Infatti, nel caso in questione i progetti **Bay** e **Switch** sono reciprocamente indifferenti, dato che due individui preferiscono strettamente **Bay**, due preferiscono strettamente **Switch** e uno è indifferente fra loro. D'altra parte, **Celestial**, **DeJaVu** e **Python** sono vicendevolmente raggiungibili attraverso preferenze dirette o catene indirette di preferenze. Più in dettaglio, i progetti **Celestial** e **DeJaVu** sono fra loro indifferenti, perché due individui preferiscono il primo, due il secondo e uno è indifferente. D'altra parte, il progetto **Python** è direttamente preferito a **Celestial** da due individui contro uno (con due indifferenti), ma **Celestial** è indifferente rispetto a **DeJaVu** e questo è preferito a **Python** da tre individui contro due.

Infine, la stampa dei gruppi determinati dall'euristica *greedy* comincia con una riga che contiene la parola chiave **Gruppi**, seguita da uno spazio e (fra parentesi tonde) la somma delle distanze di ciascun individuo dal suo rappresentante. Quindi, ogni gruppo occupa una riga di testo, che comincia con il codice fiscale dell'individuo rappresentante, seguito da uno spazio e (fra parentesi tonde) la somma delle distanze fra tale individuo e quelli del gruppo. Seguono, separati da spazi bianchi, i codici fiscali degli individui del gruppo, ordinati alfabeticamente. Le righe sono anch'esse ordinate alfabeticamente in base al codice fiscale dell'individuo che rappresenta il gruppo. Poiché può capitare che un individuo abbia la stessa distanza dai rappresentanti di più gruppi, per convenzione va assegnato al gruppo il cui rappresentante è il primo in ordine alfabetico. Analogamente, durante la scelta dei rappresentanti, se più individui, una volta aggiunti alla soluzione corrente, danno luogo allo stesso valore di distanza totale, viene scelto il primo in ordine alfabetico. Nell'esempio considerato:

Gruppi (18)

HNPFZS57R55P701N (4) JQHGVS42C59G003D

XPVQCZ02B71P804C (14) FJVWBR24L53H609K GWDTBD60C03F701L

a indicare che il costo totale della soluzione è pari a 18, dato che il primo dei due gruppi richiesti dal testo è rappresentato dall'individuo HNPFZS57R55P701N e contiene l'individuo JQHGVS42C59G003D con distanza pari a 4, mentre il secondo gruppo è rappresentato dall'individuo XPVQCZ02B71P804C e contiene gli individui FJVWBR24L53H609K e GWDTBD60C03F701L con distanza totale pari a 14 dal rappresentante. Infatti, JQHGVS42C59G003D assegna gli stessi punteggi di HNPFZS57R55P701N ai progetti DeJaVu (5) e Switch (1), mentre dà 5 punti contro 4 a Bay, 3 punti contro 4 a Python e 2 punti contro 4 a Celestial.

Chiarimenti

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