

# Laboratorio di Algoritmi

## Progetto “Pizza” (giugno 2023)

**Nota:** La scadenza del progetto è fissata per venerdì 16 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** Si vuole organizzare la gestione di una pizzeria, in particolare la definizione del suo menù, in base alle informazioni fornite da un’indagine di mercato. L’indagine riporta le preferenze di un insieme di clienti potenziali  $C$  riguardo gli ingredienti da aggiungere alla base della pizza stessa. Più precisamente, ogni intervistato ha indicato gli ingredienti che considera necessari, quelli semplicemente graditi e quelli da evitare entro un dato insieme  $I$ .

Dal punto di vista gestionale, si vuole compiere un’analisi preliminare dei dati, che ne estragga informazioni utili. A tale scopo, bisognerà determinare per ogni ingrediente il numero di clienti che lo richiedono, lo gradiscono o lo escludono, e ordinare gli ingredienti per popolarità decrescente. La popolarità sarà indicata per prima cosa dall’essere esclusi da un minor numero di clienti; a parità di questo, sarà più popolare l’ingrediente richiesto dal maggior numero di clienti; in caso di ulteriore parità, quello gradito al maggior numero di clienti. Infine, varrà l’ordine alfabetico<sup>1</sup>.

Considereremo poi il grado di esigenza dei potenziali clienti intervistati, ordinandoli per esigenza creacente. Considereremo più esigenti i clienti quelli che che impongono un maggior numero di vincoli forti (richiesta o esclusione di ingredienti, senza distinguere tra i due) e poi un maggior numero di vincoli deboli (semplice gradimento). Ancora una volta, in caso di parità, varrà l’ordine alfabetico.

Sarà anche interessante conoscere il profilo di distribuzione della popolarità degli ingredienti, vale a dire il numero di ingredienti che sono rifiutati da ciascun possibile numero di clienti.

A questo punto, è possibile cercare di determinare un menù che soddisfi tutti i clienti con il minimo numero di pizze. Siccome il problema è computazionalmente difficile, valuteremo prima una stima per difetto del valore ottimo e poi una per eccesso corrispondente alla soluzione ammissibile ottenuta con un algoritmo euristico. La stima per difetto si basa sul concetto di incompatibilità fra clienti: due clienti sono incompatibili quando non possono essere soddisfatti dalla stessa pizza, perché uno dei due richiede un ingrediente che l’altro esclude. Un gruppo di clienti incompatibili a coppie impone la produzione di una pizza diversa per ciascuno. Quindi il gruppo più numeroso di clienti incompatibili a coppie stima per difetto il numero di pizze necessario a soddisfarli tutti. Trovare tale numero è ancora un problema difficile, ma qualsiasi gruppo di clienti incompatibili lo approssima per difetto, fornendo una stima valida, anche se ancora più debole. Per ottenere questa stima partiremo dal cliente incompatibile con il massimo numero di altri clienti (come sempre, in caso di parità il primo in ordine alfabetico). Procederemo poi con un algoritmo *greedy* che aggiunge di volta in volta il cliente che sia incompatibile con tutti quelli già aggiunti al gruppo e con il massimo numero complessivo di clienti. Lo scopo del

---

<sup>1</sup>Nel progetto si intende come ordine alfabetico quello usato dalla funzione di libreria `strcmp`, che ordina anche i caratteri non alfabetici e distingue le lettere maiuscole e minuscole. Questo potenzialmente viola l’ordine alfabetico rigorosamente detto, dato che Z precede a.

criterio è di tenere disponibile il massimo numero di coppie incompatibili potenziali.

Infine, costruiremo un menù che soddisfi tutti i clienti attraverso un altro algoritmo *greedy* che cerca di determinare una pizza gradita al massimo numero di clienti non soddisfatti dalle altre. Anche questo non è un problema facile, e l'algoritmo è euristico. L'idea è aggiungere iterativamente l'ingrediente che conserva la massima platea potenziale di clienti, cioè il massimo numero di clienti che non rifiuterebbero la pizza corrente se le si aggiungesse l'ingrediente stesso. Continueremo a usare l'ordine alfabetico per risolvere i casi di parità. Una volta definita questa pizza, la si aggiunge al menù e si considerano ormai soddisfatti i clienti che la accetterebbero. Finché vi sono clienti non soddisfatti, si generano altre pizze con lo stesso algoritmo, ignorando i clienti soddisfatti, cioè non considerandoli più parte della platea potenziale. Va detto che l'insieme di pizze ottenuto potrebbe persino rivelarsi ridondante, perché le pizze definite successivamente potrebbero soddisfare tutti i clienti soddisfatti da una pizza definita in precedenza, ma per semplicità ignoriamo questo difetto.

**Il progetto** Il progetto richiede la stesura di un programma che legga le informazioni necessarie da due file di testo. Il primo fornisce l'elenco degli ingredienti disponibili, in ordine alfabetico. Si apre con il loro numero e prosegue con una riga per ciascun ingrediente. Per semplicità, ogni ingrediente è descritto da una sola parola, di al più 50 caratteri<sup>2</sup>:

10  
Acciughe  
Aglio  
Ananas  
Asparagi  
Bacon  
Banane  
Basilico  
Bresaola  
Brie  
Broccoli

Il secondo file fornisce l'elenco dei clienti intervistati, in ordine alfabetico. La prima riga del file ne fornisce il numero. Segue un blocco di quattro righe per ciascun cliente.

- la prima riga di ogni blocco riporta il nome del cliente (una singola parola di al più 50 caratteri);
- la seconda riga riporta il numero di ingredienti richiesti, seguito dal loro elenco (non ordinato alfabeticamente);
- la terza riga riporta il numero di ingredienti graditi, seguito dal loro elenco (non ordinato alfabeticamente);
- la quarta riga riporta il numero di ingredienti esclusi, seguito dal loro elenco (non ordinato alfabeticamente);

---

<sup>2</sup>Le indicazioni sulla lunghezza delle parole servono a dimensionare le stringhe. Non vanno intese come un invito a considerare  $O(1)$  il numero di possibili ingredienti, clienti o altro nelle analisi di complessità.

I tre elenchi non sono in generale ordinati alfabeticamente. Contengono solo ingredienti riportati nel primo file e sono disgiunti tra loro. Ovviamente, ci possono essere ingredienti che non sono né richiesti, né graditi, né esclusi. Per esempio:

```

5
Berardi
1 Banane
3 Bacon Asparagi Acciughe
0
Gallo
3 Aglio Ananas Banane
2 Brie Asparagi
2 Bresaola Acciughe
Lippi
4 Broccoli Brie Aglio Asparagi
4 Bresaola Acciughe Bacon Banane
0
Prelati
3 Ananas Aglio Basilico
2 Banane Bacon
1 Broccoli
Valbonesi
3 Banane Asparagi Acciughe
1 Aglio
4 Brie Broccoli Ananas Bacon

```

indica che sono stati intervistati 5 clienti. Il primo (**Berardi**) ha un solo ingrediente richiesto (**Banane**), 3 ingredienti graditi (**Bacon**, **Asparagi** e **Acciughe**) e nessun ingrediente escluso.

L'analisi gestionale richiede di stampare per prima cosa gli ingredienti in ordine di popolarità decrescente. La prima riga conterrà il numero di ingredienti che compaiono nelle indicazioni degli utenti (quindi, esclusi quelli che non vengono nominati da nessuno), seguito dalla parola chiave **ingredienti**. Ciascuna riga seguente riporterà il nome dell'ingrediente seguito dal numero di clienti che lo escludono, che lo richiedono e che lo gradiscono. Queste informazioni saranno separate da singoli spazi. Nell'esempio:

```

10 ingredienti
Banane 0 3 2
Aglio 0 3 1
Asparagi 0 2 2
Basilico 0 1 0
Ananas 1 2 0
Acciughe 1 1 2
Brie 1 1 1
Bacon 1 0 3
Bresaola 1 0 1
Broccoli 2 1 0

```

cioè tutti i 10 ingredienti sono stati nominati da qualcuno. Il più popolare sono le **Banane**, che nessuno esclude, 3 clienti richiedono e 2 gradiscono. Il meno popolare sono i **Broccoli**, che 2 clienti escludono, uno richiede e nessuno gradisce.

Quindi si stamperanno i clienti in ordine di grado di esigenza crescente. La prima riga conterrà il numero di clienti seguito dalla parola chiave **clienti**. Ciascuna riga seguente riporterà il nome del cliente seguito dal numero di vincoli forti e vincoli

deboli espressi. Queste informazioni saranno separate da singoli spazi. Nell'esempio:

```
5 clienti
Berardi 1 3
Prelati 4 2
Lippi 4 4
Gallo 5 2
Valbonesi 7 1
```

cioè il cliente meno esigente è **Berardi**, che impone solo un vincolo forte (una richiesta) e 3 deboli. Il più esigente è **Valbonesi**, che impone 7 vincoli forti (3 richieste e 4 esclusioni) e uno debole.

Infine, si stamperà la frequenza campionaria delle esclusioni, cioè il numero di ingredienti che vengono esclusi da un dato numero di clienti, limitandosi per brevità ai valori strettamente positivi. Si stamperà prima la parola chiave **esclusioni** su una riga, poi una riga per ogni valore, riportando il numero di ingredienti, seguito dalla parola chiave **per**, dal numero di clienti che li escludono e dalla parola chiave **clienti**, separate da singoli spazi. Nell'esempio:

```
esclusioni
4 per 0 clienti
5 per 1 clienti
1 per 2 clienti
```

cioè 4 ingredienti non vengono esclusi da nessun cliente, 5 ingredienti vengono esclusi da un cliente solo e un ingrediente (i **Broccoli**) viene escluso da 3 clienti. Ovviamente, la somma di questi valori è il numero totale di clienti ( $4 + 5 + 1 = 10$ ).

Dovremo poi stampare l'elenco delle coppie di clienti incompatibili, introdotto da una riga con il numero di coppie seguito dalla parola chiave **coppie incompatibili**, seguito dalle coppie di clienti stesse, in ordine alfabetico (con i clienti ordinati alfabeticamente in ciascuna coppia). Nell'esempio:

```
4 coppie incompatibili
Gallo Valbonesi
Lippi Prelati
Lippi Valbonesi
Prelati Valbonesi
```

cioè 4 coppie di clienti sono incompatibili. Per esempio, **Gallo** richiede l'**Ananas** ed esclude le **Acciughe**, mentre **Valbonesi** richiede le **Acciughe** ed esclude l'**Ananas**.

La stima per difetto del numero di pizze sarà riportata in due righe. La prima riga fornirà la parola chiave **almeno**, seguita dal numero stimato e dalla parola chiave **pizze**. La seconda riporterà i nomi dei clienti incompatibili su cui è basata la stima, in ordine alfabetico e separati da spazi singoli. Nell'esempio:

```
almeno 3 pizze
Lippi Prelati Valbonesi
```

cioè occorrono almeno 3 pizze, perché **Lippi**, **Prelati** e **Valbonesi** sono mutualmente incompatibili, come già visto sopra.

Il menù ottenuto con l'algoritmo *greedy* andrà stampato riportando per prima cosa il numero di pizze seguito dalla parola chiave **pizze**. Quindi si descriveranno le pizze nell'ordine in cui sono state ottenute. Ciascuna occuperà due righe. La prima riga conterrà l'elenco degli ingredienti in ordine alfabetico, la seconda riga

l'elenco dei clienti soddisfatti in ordine alfabetico. In entrambi i casi, i vari termini saranno separati da spazi singoli. Nell'esempio:

4 pizze

Acciughe Aglio Ananas Asparagi Bacon Banane Basilico Bresola Brie Broccoli

Berardi Lippi

Acciughe Aglio Ananas Asparagi Bacon Banane Basilico Bresola Brie

Prelati

Acciughe Aglio Asparagi Banane Basilico Bresola

Valbonesi

Aglio Ananas Asparagi Bacon Banane Basilico Brie Broccoli

Gallo

cioè l'algoritmo euristico suggerisce di preparare 4 pizze (il che rispetta la stima per difetto calcolata sopra, ma lascia aperto il problema se il numero ottimo di pizze sia 3 oppure 4). La prima pizza comprende Acciughe, Aglio, Ananas, Asparagi, Bacon, Banane, Basilico, Bresola, Brie e Broccoli, e soddisfa i clienti Berardi e Lippi.

## Chiarimenti

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