Aufgabe 2: Flohmarkt in Langdorf

Teilnahme-ID: 56860

Bearbeiter/-in dieser Aufgabe:
Christopher Besch

28. Dezember 2020

Inhaltsverzeichnis

1 Lösungsidee

Die von Donald beobachteten Spieße lassen sich wie folgt darstellen:

Schüsseln	**Früchte**
1, 4, 5	Apfel, Banane, Brombeere
3, 5, 6	Banane, Pflaume, Weintraube
1, 2, 4	Apfel, Brombeere, Erdbeere
2, 6	Erdbeere, Pflaume

Nun wird in einer Tabelle jede Schüssel mit den Früchten der Spieße, die diese Schüssel verwendeten, aufgelistet (eine Spalte pro Spieß). Zudem werden in einer weiteren Spalte alle Früchte notiert, die in Spießen vorkommen, die nicht diese Schüssel verwenden, sie werden verbotene Früchte genannt. Diese Früchte können nicht von dieser Schüssel sein.

Schüssel	**Spieße von dieser Schüssel**		**Verbotene Früchte**
1	Apfel, Banane, Brombeere	Apfel, Brombeere, Erdbeere	Banane, Weintraube, Erdbeere, Pflaume
2	Apfel, Brombeere, Erdbeere	Erdbeere, Pflaume	Apfel, Brombeere, Banane, Weintraube, Pflaume
3	Banane, Pflaume, Weintraube		Banane, Apfel, Brombeere, Erdbeere, Pflaume
4	Apfel, Banane, Brombeere	Apfel, Brombeere, Erdbeere	Banane, Weintraube, Erdbeere, Pflaume
5	Apfel, Banane, Brombeere	Banane, Pflaume, Weintraube	Apfel, Brombeere, Erdbeere, Pflaume
6	Banane, Pflaume, Weintraube	Erdbeere, Pflaume	Banane, Apfel, Brombeere, Erdbeere

Nun werden für jede Schüssel alle möglicherweise vorliegenden Früchte gesucht, diese werden im Folgenden legale Früchte genannt. An jede Schüssel liegt eine Frucht aus der Menge der legalen Früchte für diese Schüssel.

Alle legalen Früchte für eine bestimmte Schüssel müssen zwei Regeln erfüllen:

1. Die Frucht muss in allen Spießen vorkommen, die diese Schüssel benutzen.
2. Die Frucht darf für diese Schüssel nicht verboten sein.

Alle legalen Früchte werden nun markiert:

Schüssel	**Spieße von dieser Schüssel**		**Verbotene Früchte**
1	Apfel, Banane, Brombeere	Apfel, Brombeere, Erdbeere	Banane, Weintraube, Erdbeere, Pflaume
2	Apfel, Brombeere, Erdbeere	Erdbeere, Pflaume	Apfel, Brombeere, Banane, Weintraube, Pflaume
3	Banane, Pflaume, Weintraube		Banane, Apfel, Brombeere, Erdbeere, Pflaume
4	Apfel, Banane, Brombeere	Apfel, Brombeere, Erdbeere	Banane, Weintraube, Erdbeere, Pflaume
5	Apfel, Banane, Brombeere	Banane, Pflaume, Weintraube	Apfel, Brombeere, Erdbeere, Pflaume
6	Banane, Pflaume, Weintraube	Erdbeere, Pflaume	Banane, Apfel, Brombeere, Erdbeere

Die folgenden Früchte sind (von Donald) gefordert: Weintraube, Brombeere und Apfel
Es werden alle Schüsseln durchgegangen.

- Wenn alle legalen Früchte einer Schüssel gefordert sind, wird diese Schüssel ausgewählt, da ganz egal, welche legale Frucht die tatsächlich an der Schüssel liegende ist, die Schüssel eine geforderte Frucht liefert.
- Wenn keine der legalen Früchte gefordert sind, ist dies kein ausgewählter Schüssel.
- Wenn einige, aber nicht alle legalen Früchte gefordert sind, ist nicht genügend Information vorhanden, da nicht gesagt werden kann, ob der Schüssel eine ausgewählte Frucht enthält oder nicht.

Es ergibt sich die Menge aller ausgewählten Schüsseln: 1, 2 und 3
Die Schüsseln 1 und 2 liefern jeweils entweder Äpfel oder Brombeeren. Da beide Früchte gefordert sind, ist diese Ungewissheit irrelevant.
Mit dieser Methode kann in allen Fällen eine Aussage gefällt werden, ob genügend Information vorliegt und wenn dies der Fall ist, die Menge der ausgewählten Schüsseln angegeben werden.

2 Umsetzung

Die Lösungsidee wird in C++ implementiert. Als erster Schritt wird in der Funktion **read_file** die Eingabedatei gelesen. Hierbei wird überprüft, ob die Eingabedatei dem gegebenen Format entspricht, wenn nicht wird das Programm abgebrochen.

Die Fruchtnamen werden mithilfe einer Instanz der Klasse **LookupTable** aufsteigend und anfangend bei 0 in Glanzzahlen, Frucht IDs, übersetzt. Dadurch ist die Anzahl an verschieden Früchten nicht limitiert. Ein weiterer Vorteil ist, dass da die ID aufsteigend und bei 0 anfangend verteilt werden, die IDs als Indizes in Bit Fields verwendet werden können. So liegt für jede Frucht ein Bit/Bool vor, ist er true, ist diese Frucht enthalten, ist sie false, ist sie nicht enthalten. Diese kompakte Datenstruktur ist ressourcenschonender im Vergleich zu einem std::vector<std::string> mit den Namen aller enthaltenden Früchten.

Die Namen für die Schüsseln werden auf die gleiche Weise behandelt. Zwar ist angegeben, dass alle Schüsseln nummeriert sind, es ist allerdings nicht klar, wie die Nummern verteilt werden, daher ergibt es Sinn, die Namen der Schüsseln als Strings zu behandeln und ebenfalls eine Instanz von **LookupTable** zu verwenden. So können auch für die Schüsseln Bit Fields verwendet werden. Ein zusätzlicher Vorteil ist, dass so jegliche Namen für die Schüsseln erlaubt sind.

Es werden Instanzen der Klasse **Skewer** erstellt, diese enthalten die Menge der Früchte auf einem Spieß und die Menge der Schüsseln, von denen die Früchte genommen sind. Hierbei werden, wie bereits erwähnt, Bit Fields verwendet.

3 Beispiele

4 Quellcode