Problem B
Szarlotka z lodami

Sformułujmy to zadanie w języku grafów. Niech V oznacza zbiór wszystkich kawałków ciasta, a W zbiór wszystkich gałek lodów, $|V|=|W|$. Zbudujemy graf dwudzielny o wierzchołkach $V \cup W$. Krawędź pomiędzy kawałkiem ciasta $v \in V$ i gałką lodów $w \in W$ tworzymy jeśli dana kombinacja jest możliwa. Waga krawędzi to cena danej kombinacji.

Skojarzeniem w grafie nazywamy podzbiór zbioru krawędzi, taki że żadne dwie krawędzie nie mają wspólnego wierzchołka. Skojarzenie doskonałe to takie, w którym każdy wierzchołek jest incydentny z jakąś krawędzią tego skojarzenia. Skojarzenie doskonałe w naszym grafie dwudzielnym odpowiada pewnemu przyporządkowaniu wszystkich kawałków ciasta do wzystkich gałek lodów. Z treści zadania wiadomo, że co najmniej jedno takie skojarzenie doskonałe istnieje.

Waga skojarzenia M (oznaczmy ją jako $\sigma(M)$) to suma wag wszystkich krawędzi wchodzących w jego skład. W zadaniu chodzi o policzenie najmniejszej i największej wagi skojarzenia doskonałego w grafie dwudzielnym. Jest to znany problem w informatyce, nazywany *problemem przydziału*. Występuje on w wielu praktycznych zastosowaniach, dlatego wymyślono wiele algorytmów do jego rozwiązania. Najszybsze znane algorytmy działają w czasie $O(n^3)$, na przykład *algorytm węgierski* lub *algorytm kolejnych najkrótszych ścieżek*. Nasze rozwiązanie będzie opierać się na drugim z nich, który dla prostoty zastosujemy w wersji działającej w czasie $O(n^4)$, co okaże się zupełnie wystarczające.

Definicja 1 *Niech $G = (V, W, E, M)$ oznacza graf dwudzielny o zbiorach wierzchołków V i W, krawędziach E i skojarzeniu M.* **Ścieżką powiększającą** *w grafie G nazywamy ścieżkę od wierzchołka nieskojarzonego $v \in V$ do wierzchołka nieskojarzonego $w \in W$ taką, że jej kolejne krawędzie na przemian raz należą do skojarzenia M, raz nie.* **Waga ścieżki powiększającej**, *$\sigma(\mathcal{P})$ to suma wag krawędzi ścieżki nie należących do M minus suma wag krawędzi ścieżki należących do M.*

Zauważmy, że pierwsza i ostatnia krawędź na ścieżce powiększającej nie należy do skojarzenia. Gdy na ścieżce powiększającej zamienimy dla każdej krawędzi jej przynależność do M, to uzyskamy nowe skojarzenie M', zawierające o jedną krawędź więcej niż M. Obliczając wagę ścieżki powiększającej, sumujemy wagi jej krawędzi, ale krawędzie należące do skojarzenia bierzemy z przeciwnym znakiem. Zauważmy, że jeśli ścieżka $\mathcal{P}$ powiększa skojarzenie M do skojarzenia M', to $\sigma(M')=\sigma(M) + \sigma(\mathcal{P})$.

Algorytm kolejnych najkrótszych ścieżek opiera się na następującym twierdzeniu:

Twierdzenie 1 *Niech M będzie skojarzeniem w grafie dwudzielnym $G=(V, W, E, M)$ o najmniejszej wadze spośród skojarzeń o liczności k. Niech $\mathcal{P}$ będzie najkrótszą ścieżką powiększającą. Wtedy skojarzenie M', powstałe poprzez zastosowanie ścieżki powiększającej $\mathcal{P}$ jest skojarzeniem o najmniejszej wadze spośród skojarzeń o liczności $k + 1$.*

Algorytm kolejnych najkrótszych ścieżek jest następujący:

```
Rozpocznij z pustym skojarzeniem M
Powtarzaj:
   Znajdź najkrótszą ścieżkę powiększającą
   Jeśli znaleziono ścieżkę, powiększ M zgodnie ze znalezioną ścieżka
   w przeciwnym przypadku KONIEC
```

Zgodnie z twierdzeniem 1 po zakończeniu działania, M będzie skojarzeniem doskonałym o najmniejszej wadze (o ile skojarzenie doskonałe w ogóle istnieje).

W celu znalezienia najkrótszej ścieżki powiększającej, zmodyfikujemy nieco graf G:
- krawędziom nie należącym do skojarzenia M nadajemy kierunek od V do W
- krawędziom należącym do skojarzenia nadajemy kierunek od W do V i zmieniamy wagę na

przeciwną
- dodajemy wierzchołek startowy s, od którego prowadzą krawędzie o wadze 0 do wszystkich wierzchołków nieskojarzonych w zbiorze V

W tak zmodyfikowanym grafie wystarczy znaleźć najkrótszą ścieżkę od s do wierzchołka nieskojarzonego $w \in W$ - w oczywisty sposób odpowiada ona najkrótszej ścieżce powiększającej. Zauważmy, że w żadnym kroku algorytmu graf zmodyfikowany nie zawiera cykli o ujemnej wadze. Gdyby tak było, można by było ten cykl zastosować tak jak ścieżkę powiększającą, uzyskując skojarzenie o tej samej liczbie krawędzi, ale mniejszej wadze. Ten fakt uprawnia nas do zastosowania algorytmu Bellmana-Forda do znalezienia najkrótszej ścieżki. Algorytmu Dijktry nie można zastosować ze względu na występowanie ujemnych wag krawędzi. Okazuje się, że z ujemnymi wagami można sobie poradzić, dodając tzw. *potencjał* do wierzchołków, ale nie będziemy się tym zajmować.

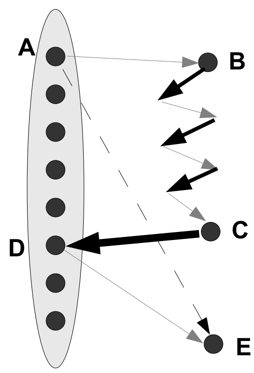
Algorytm kolejnych najkrótszych ścieżek ma następującą własność, którą wykorzystamy w naszym rozwiązaniu.

Fakt 1 *W algorytmie kolejnych najkrótszych ścieżek wagi kolejnych ścieżek tworzą ciąg niemalejący.*

Największa możliwa ilość wierzchołków ($|V \cup W|$) to 10000 (5000 kawałków ciasta i 5000 gałek lodów). Zastosowanie algorytmu o złożoności czasowej rzędu n^3, nie mówiąc już o n^4 nie jest dobrym pomysłem. Spróbujmy wykorzystać fakt, że maksymalnie mamy tylko 50 rodzajów ciasta i 50 smaków lodów.

Grupą nazwijmy zbiór wszystkich wierzchołków tego samego typu, tzn. wszystkich kawałków ciasta tego samego rodzaju lub wszystkich gałek lodów o tym samym smaku. Zastanówmy się, czy w algorytmie kolejnych najkrótszych ścieżek istnieją ścieżki wielokrotnie przechodzące przez jedną grupę.

Na poniższym schemacie narysowany jest przypadek „wrócenia" ścieżki do tej samej grupy. Ścieżka po raz pierwszy trafia do grupy w wierzchołku A, potem wraca w wierzchołku D. Ścieżki należące do skojarzenia są pogrubione. Niech $|XY|$ oznacza długość ścieżki od X do Y. Wierzchołki A i D są z tej samej grupy, zatem $|AB| = |DB|$ oraz $|AE| = |DE|$. Gdyby trasa $A \to B \to \ldots \to C \to D \to E$ była krótsza niż $A \to E$, to ścieżka $A \to B \to \ldots \to C \to D$ musiałaby mieć ujemną długość. Długość tej ścieżki jest równa długości cyklu $D \to B \to \ldots \to C \to D$. Jak wiemy, ten graf nie zawiera ujemnych cykli, zatem ścieżka $A \to B \to \ldots \to C \to D$ nie może mieć ujemnej długości. Stąd wniosek, że możemy pójść od A bezpośrednio do E, eliminując wracanie do tej samej grupy.



Jeśli zatem możemy ograniczyć się tylko do ścieżek przechodzących co najwyżej raz przez daną grupę, to możemy szukać najkrótszej ścieżki w grafie *skompresowanym*, w którym wszystkie wierzchołki z danej grupy połączone są w jeden. Innymi słowy wierzchołkami zamiast poszczególnych kawałków ciasta są rodzaje ciast, a zamiast poszczególnych gałek lodów są smaki lodów.

Do pamiętania skojarzenia potrzebne będą następujące struktury danych:
- *matched* - tablica dwuwymiarowa, element $matched[i, j]$ określa ile ciastek z grupy i jest skojarzonych z lodami z grupy j
- mV, mW - tablice jednowymiarowe, $mV[i]$ określa, ile ciastek z grupy i jest skojarzonych, analogicznie mW

Po znalezieniu ścieżki w grafie skompresowanym, możemy jej użyć do powiększenia skojarzenia o 1 w grafie nieskompresowanym. Ponieważ jednak, zgodnie z faktem 1, następna ścieżka nie będzie miała mniejszej wagi, to znalezioną ścieżkę możemy wykorzystać wielokrotnie. Dlatego w naszym algorytmie, po znalezieniu ścieżki powiększającej wpierw przejdziemy przez nią, żeby obliczyć, ile razy można ją wykorzystać. W drugim przejściu ścieżka zostanie wykorzystana.

Tak skonstruowany algorytm dla znajdowania minimalnego zysku łatwo jest wykorzystać do znalezienia zysku maksymalnego. Wystarczy zamienić wszystkie ceny na przeciwne. Żeby nie było cen ujemnch, dodajemy do każdej 2000. Przy tak zmienionych wagach obliczamy zysk minimalny, po czym odejmujemy $N * 2000$, gdzie N to ilość wszystkich sprzedanych deserów, następnie zmieniamy znak na przeciwny.

Testy

b0 – test z treści zadania, dwa zestawy
b1 – 3 zestawy, łączna ilość deserów równa 10
b2 – 5 zestawów, łączna ilość deserów równa 100
b3 – 50 zestawów, łączna ilość deserów równa 100
b4 – 30 zestawów, łączna ilość deserów równa 1000
b5 – 100 zestawów, łączna ilość deserów równa 3000, około 50% możliwych krawedzi nie istnieje
b6 – 10 zestawów 50x50 typów, ilości 10..100, ceny 0.01..10.00, 10% możliwych krawędzi nie istnieje
b7 – 10 zestawów 50x50 typów, ilości tylko 100, ceny 0.01..100.0, graf pełny dwudzielny
b8 – 10 zestawów 50x50 typów, ilości 90..100, ceny 0.01..100.0, 50% możliwych krawędzi nie istnieje
b9 – 10 zestawów 50x50 typów, ilości 50..100, ceny tylko 0.01 lub 0.02, 10% możliwych krawędzi nie istnieje
b10 - 10 zestawów 50x50 typów, ilości 1..50, ceny 50.00..60.00, graf pełny dwudzielny