



Bankrupt Baker

Link submit:

https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=2283

Solution:

C++	https://ideone.com/a1br17
Java	https://ideone.com/GaSZ1L
Python	https://ideone.com/0b7VtE

Tóm tắt đề:

Wolfgang Puck có một bộ sưu tập các công thức nấu ăn bánh. Chúng được phân thành ra để trong các kệ bướm khác nhau phụ thuộc vào loại bánh. Mặc dù Wolfgang có các cửa hàng nhượng quyền thương hiệu trên khắp thế giới, anh đang trong giai đoạn khó khăn và đang phải vật lộn để mua nguyên liệu cho bánh. Anh ta có thể tạo ra bánh gì với ngân sách nhỏ của mình?

Input:

Dòng đầu tiên là số nguyên T – số lượng kệ bướm, tiếp theo là thông tin của các kệ đó.

Mỗi kệ bắt đầu bằng tên *title* là tên của nó, tiếp theo là một dòng chứa ba số nguyên dương m n b ($1 \leq n, m \leq 100, 1 \leq b \leq 10^6$) với b là ngân sách của Wolfgang tính theo USD.

Tiếp theo là m dòng, mỗi dòng có cấu trúc *ingredient* c mô tả một đơn vị thành phần *ingredient* sẽ có giá là c USD ($0 \leq c \leq 5000$).

Tiếp theo là n công thức, mỗi công thức sẽ bắt đầu bằng một dòng chứa tên công thức đó, dòng tiếp theo là một số k – số lượng thành phần cần để nấu theo công thức.

k dòng tiếp theo, mỗi dòng có dạng *requirement* x , tức sẽ cần x đơn vị thành phần *requirement*.

Output:

Với mỗi kệ bướm, in ra danh sách các công thức mà Wolfgang có thể chế biến được trong giới hạn ngân sách cho phép của mình, các công thức in theo thứ tự tăng dần của chi phí sản xuất, nếu cùng giá thì in theo thứ tự tăng dần theo tên. Nếu trong kệ bướm đó không thể chế biến bất kì công thức nào thì in ra "Too expensive!". In ra một dòng trống sau mỗi kệ bướm.

Ví dụ:

<pre>2 My Favourite Cheesecake 8 3 100 sugar 4 water 0 lemonjuice 3 creamcheese 20 vanilla 5 egg 5 cream 10 strawberry 5 Strawberry Whipped Cream 2 cream 5 strawberry 3 Scrumptious Caramel Topping 3 sugar 6 water 3 lemonjuice 1 Secret Cheesecake Base 5 creamcheese 3 sugar 5 vanilla 1 egg 6 cream 1 Million Dollar Cakes 3 1 999999 costlyflour 500 gold 4500 diamond 5000 Display Cake - Do Not Eat! 3 costlyflour 100 gold 100 diamond 100</pre>	<pre>MY FAVOURITE CHEESECAKE Scrumptious Caramel Topping Strawberry Whipped Cream MILLION DOLLAR CAKES Too expensive!</pre>
---	--

Hướng dẫn giải:

Với mỗi kẹp bướm:

- Ta sử dụng một map (với C++, tương ứng là TreeSet/HashSet với Java hoặc dict với Python) để lưu lại danh sách các thành phần cùng với đơn giá của chúng. Ngoài ra, sử dụng một mảng có kiểu dữ liệu là pair<int, string> để lưu lại đơn giá và tên của các công thức trong kẹp bướm (với Java và python thì có thể sử dụng các cách tương ứng).

- Sau đó, với mỗi công thức chế biến, ta sẽ tính tổng chi phí để thực hiện nó dựa theo số lượng và đơn giá đã có. Sau khi tính được đơn giá của công thức, nếu đơn giá đó nằm trong giới hạn b cho phép thì ta thêm nó vào danh sách các đơn giá-công thức đã có.
- Lúc này sẽ có hai trường hợp: danh sách rỗng thì in ra "Too expensive!". Ngược lại thì ta sắp xếp lại tăng dần theo giá, cùng giá thì tăng dần theo tên. Cuối cùng là in ra danh sách các công thức theo đúng thứ tự.

Độ phức tạp: $O(T * \sum(k_i) * \log m + n * \log n)$, trong đó, T là số lượng kẹp bướm, và để tính được đơn giá của tất cả các công thức thì ta phải duyệt qua tất cả các thành phần, với mỗi thành phần thì tìm kiếm nó trong map, như vậy chi phí là $O(\sum(k_i) * \log(m))$ (với k_i là số thành phần trong công thức thứ i), chi phí để sắp xếp trong một tệp là $O(n * \log n)$

Big-O Coding