АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
99 БАЛОНА

The solution is based on graph expansion and dijkstra.

We need to keep track of what conflicts we have just passed, so we know not to enter countries that won't let us in.

Each vertex is expanded into (K-1) * (C+1) vertices -- to track the K-1 last conflicts, plus one more to denote nothing.

Решението се базира на дейкстра с разширение на графа.

Понеже има ограничение за държавите които може да се посещават в зависимост от последните посетени, трябва да пазим достатъчно информация за да преценяваме дали може да посетим конкретна държава на конкретен ход. Наивния подход е да пазим K-те последни държави в пътя. На всеки ход проверяваме дали новата евентуална държава е в конфликт с някоя от предишните K.

За съжаление този подход няма да работи за всички тестове, защото броя върхове се вдига до N^K (списък от K върха). Достатъчно е да пазим само в кои конфликти сме били последните K-1 хода, и в кой връх сме в момента. Така броя върхове в разширения граф е $N * (C+1) ^ {(K-1)}$. C+1 се получава защото трябва да добавим нещо като виртуален конфликт, който представлява липса на конфликт (за да попълним "опашката" в началото на пътя с несъществуващи конфликти).

Сега ще определим кои са върховете и ребрата на разширения граф.

$(c_1, c_2, ... c_{k-1}, n)$ е връх, където

* нека c_k да е конфликта, в който участва връх n

* c_i е конфликт $\in \{0, 1, ..., c\}$, където с 0 се отбелязва липсата на конфликт, $i \in \{1, k\}$, $c_k != 0$

* $c_i == 0$, само ако $i == 1$ или $c_{i-1} == 0$ (само първите няколко конфликта може да липсват)

* ако $i, j \in \{1, k\}$, $i != j$, и $c_i != 0$ и $c_j != 0$, то $c_i != c_j$ (няма повторение на конфликти)

Ребро от $(c_1, c_2, ... c_{k-1}, n)$ до $(c'_1, c'_2, ..., c'_{k-1}, n')$ има тогава и само тогава когато:

* приемаме, че c_k е конфликта в който участва n, а c'_k е конфликта в който участва n'

* $c_i == c'_{i-1}$, за i от 2 до k

От имплементационна гледна точка, за да пазим по-ефективно c_1 до c_{k-1} за всеки връх може да използваме по 4 бита за c_i, и така всички c_iсе събират в unsigned int. Отделно върха на графа се събира в int, следователно връх в разширения граф се събира в 64 битово цяло число -- по-лесно се вкарва в хеш таблици и приоритетни опашки.