

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА АВИОЛИНИИ

Пълното решение на задачата изключва явното създаване на граф с всички полети, в който да се решава задача за намиране на най-къси разстояния, тъй като броят на ребрата може да расте експоненциално с нарастването на броя операции. За да решим задачата, ще построим нов граф, в който, освен началните N върха, ще има още $O(M)$ върха и общо $O(N+M)$ ребра. При това, дължината на най-късия път от връх 1 (столицата) до всеки от началните N върха ще бъде същата, както и в графа с всички полети, който би се създал след изпълнение на всички операции (да наречем този граф „първичен“). Новият граф строим по следния начин:

- за всеки връх v от първичния граф, т.е. за всеки град, да създадем два върха в новия граф, които да означим с in_v и $active_v$;
- при всяка операция от тип 1, с параметри A , B и C , в новия граф се създава ребро от върха $active_A$ към върха in_B с дължина C .
- при всяка операция от тип 2, с параметри A , B и C , в новия граф се създава само едно ребро от върха $active_B$ до върха $active_A$ с дължина C . Да си спомним, че при операция от тип 2 в първичния граф (който ние не строим наистина) трябваше да се създадат ребра от върха B до всички върхове, до които има пряко ребро от A , като при това дължината на тези ребра трябваше да е равна на дължината на реброто от A до съответния връх плюс C . В новия граф с прекарването на ребро с дължина C от върха $active_B$ до върха $active_A$ (и още малко допълнителни работи, които са описани по-долу) се получава път от върха B до всички върхове, до които има пряко ребро от A със същата дължина както са ребрата от B до същите върхове в първичния граф. Като че ли успяхме да заменим многото нови ребра, които могат да възникнат при изпълнение на операция 2, само с едно ребро. Има, обаче, един проблем: ако сега се появи операция от тип 1 с начален връх A и краен X , то построявайки ребро от върха $active_A$ към върха in_X ще получим, че има път от върха B до върха X , което не е така. Ето защо създаваме нов връх $active_A$, на стария $active_A$ даваме друго име (или номер) и от новия $active_A$ до стария построяваме ребро с дължина 0. Всички ребра, които влизат и излизат от стария $active_A$ си остават така, а от новия към момента излиза само едно. От следващите операции ще се използва новият $active_A$ и ще започнат да излизат и влизат ребра от и към него. По този начин, ако се изпълни операция 1 с начало новия връх $active_A$ и край някакъв връх in_x , то няма да се получи път от $active_B$ до in_x .

След изпълнението на всички операции, се построяват ребра с дължина 0 от in_v до $active_v$ за всяко v (не е правилно тези ребра да се построят в началото, тъй като, както видяхме, върховете $active_v$ могат да се променят).

В така построения граф се търсят най-късите пътища от in_1 до всички върхове от тип in .

Ако се използва алгоритъм на Форд-Белман, то ще се получат 15 точки. Ако се използва алгоритъм на Дейкстра, то може да се достигне до 55 точки, тъй като построеният граф може да съдържа ребра с отрицателна дължина.

За да се отървем от ребрата с отрицателна дължина в новия граф, да си спомним, че в първичния граф всички ребра са със строго положителна дължина.

Да разгледаме някой връх от новия граф, който не е от тип in . Нека най-малката дължина на влизащо в него ребро е d . Ние можем да извадим d от дължините на всички влизащи във върха ребра, а да добавим d към дължините на всички излизащи от върха ребра. При тази операция всеки път, който минава през този връх, без да завършва в него, няма да измени дължината си (да си спомним, че ние ще търсим минималните дължини на пътища от in_1 до всички върхове от тип in). След тази операция, в съответния връх вече не влизат отрицателни ребра. Така че, с помощта на такива операции, ние можем да премахваме от графа ребра с отрицателна дължина, като в същото време, обаче, може да се случи да добавим нови ребра с отрицателна дължина.

Тъй като, в началото, всички ребра, които влизат във върхове от тип *in* са положителни (такива ребра се получават само в резултат на операции от тип 1, при които по условие $C > 0$), а също така в графа не съществуват цикли, които да не съдържат нито един връх от тип *in*, то след краен брой операции в графа ще останат само неотрицателни ребра.

Това дава следната идея за решаване на задачата. Премахваме всички върхове от тип *in*, при което се получава ацикличен граф. Сортираме топологично този граф. След това изпълняваме операциите по промяна на дължините на ребрата на графа, описани по-горе. Тъй като графът е ацикличен, то след изпълнението на всички операции, в него няма да има ребра с отрицателна дължина. Може да се приложи алгоритъма на Дейкстра (при всички операции по преобразуването на графа дължините на пътищата от столицата до различните градове се запазват).

Сложността на този алгоритъм е $O((N+M)\log(N+M))$.