**Анализ**

Задачата е разделена на много подзадачи, което позволява различни частични решения да хванат подходящи точки, като няма как да се хващат произволни точки. Очевидно условието задава претеглен граф (забележете, че теглата на ребрата са доста малки) и се иска да се намери броя пътища с дължина, деляща се на определено число. Разбира се, понеже графът е неориентиран, е зададена максимална дължина на тези пътища, иначе отговорът би бил безкрайност. Под дължина на път разбираме сбора от теглата на ребрата в него. Задачата изисква да се направят поредица от наблюдения и алгоритми, които би трябвало да се знаят от всички състезатели в тази група. Оказва се, че в зависимост от ограниченията има различно решение, което е оптимално. Затова тази задача е от по-редките, за които трябва да се комбинират различни решения за 100 точки.

Първата подзадача е за 5 точки. Не се очаква да се направят никакви наблюдения – предвид ограниченията е достатъчно да се направи просто пълно изчерпване по отговора. Пуска се DFS от всеки връх и се обхождат всички възможни пътища, като се гледа кои са с дължина, деляща се на ***D*** и ≤ на ***K***. Очакваната сложност е $O(KN^{K})$. Реално решението върви доста по-бързо от очакваното.

Втората подзадача е за 30 точки. Всъщност тя е много важна – това e едно от решенията, без които няма как да се изкарат пълен брой точки. Ще решим задачата чрез метода динамично оптимиране. Единственото, което ни трябва като информация за текущия път е до кой връх сме стигнали и колко път сме изминали. Така стейтът може да ни е $dp\left[vr\right]\left[path\right]$, където $vr$е номерът на върха, а $path$ **-** дължината на изминатия път. Очевидно $dp\left[vr\right]\left[path\right]=\sum\_{i=1}^{N}dp\left[i\right][path-weight(i,vr)]$ като се сумира само за $i$-тата, които имат ребро до връх $vr$ и когато $path-weight(i,vr)$ е неотрицателно (с $weight(i,vr)$ е означено теглото на реброто между върховете $i$ и $vr$). Понеже този алгоритъм е сравнително прост, това се оказва по-лесната част от задачата и са отделени само 30 точки. Очакваната сложност за тази подзадача е $O\left(KM\right)$.

Третата подзадача е за 10 точки. Сега вече минаваме по същество към останалата част от задачата. Тук ще разширяваме графа като добавим нови върхове и ребра. Понеже в общия случай не можем да смятаме лесно брой пътища с определена дължина в претеглен граф, ще го направим непретеглен. Нека имаме някакво насочено ребро (ребрата са неориентирани, но за удобство тук ще ги разглеждаме като двойка ориентирани) $(x,y)$, което е ребро от $x$ към $y$ с тегло $t$. Ще въведем следните ориентирани ребра: $\left(x,v\_{1}\right), \left(v\_{1},v\_{2}\right),\left(v\_{2},v\_{3}\right),…,\left(v\_{t-2},v\_{t-1}\right),\left(v\_{t-1},y\right)$, които са с тегло 1. Ориентираният път $x,v\_{1}, v\_{2},v\_{3},…,v\_{t-2},v\_{t-1},y$ e с дължина $t$, т.е. той изцяло заменя реброто $(x,y)$. Така ако премахнем това ребро бройката на пътища ще е почти същата като в началния граф. Единствено не трябва да гледаме пътища, които свършват във нов връх $v\_{i}$ или започват от такъв връх. Това означава, че ако заменим всички ребра от началния граф по указания начин, ще получим ориентиран граф, който е непретеглен (всички ребра са с тегло 1). Нека разглеждаме неговата матрица на съседство $A$. Сега броят пътища с определена дължина $p$ от началния граф са тези, които почват от оригинален връх и завършват в някакъв оригинален и тяхната бройка можем да намерим, като вдигнем матрицата на степен $p$, т.е. гледайки матрицата $A^{p}$. Понеже в тази подзадача ***D*** = ***K*** e достатъчно да сметнем нужния брой като гледаме матрицата $A^{K}$. Това можем да направим като използваме алгоритъма за бързо повдигане в степен, тъй като свойствата на степените се запазват за матриците. Нека сметнем броя върхове в новия граф - за всяко ребро с някакво тегло $t$ от оригиналния граф (ние сме ги удвоили, защото заменихме всяко неориентирано с две ориентирани) сме добавили $t-1$ върха, което означава, че всички върхове са станали $2\*\sum\_{i=1}^{M}\left(t\_{i}-1\right)+N=2\*\sum\_{i=1}^{M}t\_{i}+N-2M$, което не е толкова много, заради ограниченията на подзадачата. Очакваната сложност е $O((\sum\_{i=1}^{M}t\_{i})^{3}log\_{2}K)$ ($N-2M$не участва в сложността, защото то е отрицателно).

Четвъртата подзадача е за 20 точки. Почти всичко е като предната задача единствено имаме проблем, че ***D*** ≠ ***K***. Ако просто пресметнем отговора като смятаме $A^{D}, A^{2D}, …, A^{\left[\frac{K}{D}\right]\*D}$ ще получим доста лоша сложност дори и да го реализираме възможно най-добре – $O\left(\left(\sum\_{i=1}^{M}t\_{i}\right)^{3}\*\left(\left[\frac{K}{D}\right]+log\_{2}D\right)\right)$. Матриците $A^{D}, A^{2D}, …, A^{\left[\frac{K}{D}\right]\*D}$ можем да представим и по следния начин $(A^{D})^{1}, (A^{D})^{2}, …, (A^{D})^{\left[\frac{K}{D}\right]}$, това означава, че е удобно първо да повдигнем матрицата $A$ на степен $D$. Нека $A^{D}=B$ и искаме да намерим начин, така че да бързо да сметнем сбора на броя пътища в $B^{1}, B^{2}, …, B^{\left[\frac{K}{D}\right]}$ от всеки връх от оригиналния граф до всеки връх от оригиналния граф. Ако използваме алгоритъма за бързо повдигане на степен дори и да сумираме на всяка стъпка ще успеем да сметнем само малка част от отговора. Затова ще въведем нулев ред със свойството, че на матрица $B^{s}$, $B^{s}\left[0\right]\left[j\right]=\sum\_{r=1}^{s}\sum\_{i=1}^{N}B^{r}[i][j]$ (смисълът е, че в $B^{s}\left[0\right]\left[j\right]$ се пазят броя пътища от произволен връх $i\leq N$до $j$ с дължина $\leq s$), като в частност $B^{s}\left[0\right]\left[0\right]=1$. Забележете, че в така написаната сума за колона $j$ не се пазят броя пътища, които започват от добавен връх – както казахме те не ни и трябват. Можем да се убедим, че стига в матрицата $B^{1}$ да се попълни както трябва този ред, което може да стане просто със смятане на сумата на всяка колона до $N$ ред, извършвайки умножение на матрица $B^{s}$ със матрица $B^{r}$, при което се получава матрица $B^{r+s}$, се запазва свойството за нулевия ред. Това е така, защото $B^{s+r}\left[0\right]\left[j\right]=\sum\_{}^{}B^{s}\left[0\right]\left[k\right]\*B^{r}[k][j]$ ($k$ минава всички върхове и нула, включително и новите), като $B^{s}\left[0\right]\left[k\right]$ са броя пътища с дължина по-малка или равна на $s$, от който и да е връх $i\leq N$ до $k$, умножени по $B^{r}[k][j]$, което е броя пътища с точна дължина $r$ от връх $k$ до връх $j$, което очевидно ще даде броя пътища с дължина $>r$ и $\leq r+s$, които са от произволен връх $i$ до $j$, които са минали през върха $k$, когато са били на дължина $r$. Понеже сумираме за всяко $k$ (включително и 0, при което просто ще се добавят пътищата с дължина $\leq r$), ще се изчерпат всички възможности и наистина $B^{s+r}\left[0\right]\left[j\right]$ ще даде търсеното от нас число за всяко $j$. Това запазване на свойството ни позволява да нямаме притеснения и стига да сме въвели правилен нулев ред да извършим спокойно алгоритъма за бързо повдигане на степен. Сега сложността става $O\left(\left(\sum\_{i=1}^{M}t\_{i}\right)^{3}\left(log\_{2}D+log\_{2}\left[\frac{K}{D}\right]\right)\right)$.

Петата подзадача е за 15 точки. Тя е за хората, които не са успели да измислят предното, но са направили важно съобразяване, с което се намалят броя върхове в новия граф значително. Нека разглеждаме всички оригинални влизащи ребра в даден връх $i$: $\left(v\_{1},i,t\_{1}\right),…, (v\_{q},i,t\_{q})$ (третото число в скобите е теглото на реброто, като както казахме неориентираните ребра са превърнати в двойка ориентирани). По схемата, която ще използвахме досега щяхме да добавим $\left(v\_{1},v\_{1,1}\right), \left(v\_{1,1},v\_{1,2}\right),…,\left(v\_{1,t\_{1}-2},v\_{t\_{1}-1}\right),\left(v\_{1,t\_{1}-1},i\right)$ за първото ребро, $\left(v\_{2},v\_{2,1}\right), \left(v\_{2,1},v\_{2,2}\right),…,\left(v\_{2,t\_{2}-2},v\_{t\_{2}-1}\right),\left(v\_{2,t\_{2}-1},i\right)$ за второто ребро, ... $\left(v\_{q},v\_{q,1}\right), \left(v\_{q,1},v\_{q,2}\right),…,\left(v\_{q,t\_{q}-2},v\_{t\_{q}-1}\right),\left(v\_{q,t\_{q}-1},i\right)$ за $q$–тото ребро. В действителност правим много излишни върхове. Като се замислим, ако вече сме направили някакви върхове за дължина $t$ можем да ползваме част от тях за друга дължина $t'$ и по-точно да я намалим с $t$ и да направим $t^{'}-t$ нови върха, които да влизат във верижката с $t$ върха, ако $t^{'}>t$, иначе можем да вземем верижката с дължина $t'$ и дори да не добавяме върхове! Ако тази логика я обобщим ще видим, че след като направим верижката с дължина $t\_{max}$, за всички останали ще добавим едно ребро към някаква нейна суфиксна част, което ще ни осигури нужния път, който да замени някакво текущо входящо ребро. Формално казано ако верижката за $t\_{max}$ (построена за ребро с теглото $t\_{max}$ за някакъв връх $i$) е $\left(v\_{max},v\_{max,1}\right), \left(v\_{max,1},v\_{max,2}\right),…,\left(v\_{max,t\_{max}-2},v\_{max,t\_{max}-1}\right),\left(v\_{max,t\_{max}-1},i\right)$ за ребро от $v\_{h}$ имаме: $\left(v\_{h},v\_{h,t\_{max}-1-(t\_{h}-1),1}\right), \left(v\_{h,t\_{max}-1-(t\_{h}-1)},v\_{h,t\_{max}-1-(t\_{h}-2)}\right),…,\left(v\_{h,t\_{max}-1},i\right)$ като верижка (ако случайно реброто е с дължина 1 в оригиналния граф, не се добавят нови върхове и ребра в графа). Сега сме сигурни, че добавените върхове няма да надминат $t\_{max}\*N$, което е много по-добре предвид, че в ограниченията максималното ребро е с дължина до 10. Сложността сега става (трябва да имаме предвид, че в тази подзадача ***D*** *=* ***K***) e $O((t\_{max}\*N)^{3}log\_{2}K)$.

Последната подзадача е за 20 точки. Тя комбинира направеното наблюдение, което намалява значително броя върхове на новия граф, както и схемата да смятаме отговора при ***D*** *≠* ***K***. Окончателната сложност тук е $O\left((t\_{max}\*N)^{3}\left(log\_{2}D+log\_{2}\left[\frac{K}{D}\right]\right)\right)$.

Задачата е интересна с това, че трябва да се напишат две решения, за да се получат 100 точки. Решението с динамично оптимиране е еквивалентно на разширяване на графа, като за всеки връх се добавя информация за изминатия път. Това решение е по-добро при повече върхове, но при по-малко ***K***. Докато при второто решение разширяването е за сметка на теглата на ребрата (което зависи от броя върхове), но почти без значение на ***K***. Резултата, който използвахме за смятане на нужния сбор, може да се обобщи ако ни трябва да сметнем бързо $(A^{s})^{1}+ (A^{s})^{2}+ …+(A^{s})^{r}$, което няма как да стане с формулата за геометрична прогресия, защото имаме матрици, но въпреки това с въвеждането на този допълнителен нулев ред се извършва ефикасно.

*Автор: Илиян Йорданов*