**Подзадача 1.** За всяка заявка се симулира движението на Теди в лабиринта – започва се от началото и се движим клетка по клетка според посоките. Важно е по време на симулацията за всяка премината клетка да се отбелязва, че вече е посетена. Това дава възможност да се засече повторението на клетка и съответно да се намери, че няма изход от лабиринта. Накрая трябва по обратния път на посещаването или с ново минаване да зачистим информацията за посетените клетки.

От тук нататък за всички подзадачи ще трябва да се направят няколко наблюдения. Да видим какви възможности имаме за заявките.

* Ако от началната клетка не минаваме през променената
* Ако от началната клетка стигаме до променената и тя води до цикъл (или до променената)
* Ако от началната клетка стигаме до променената и тя води извън лабиринта

|  |  |
| --- | --- |
| v<<v<^v>^ | v<<v<^>>^ |

Сега да видим как може да разберем до къде се стига от променената клетка. Имаме промяна и нямаме предварително запазена информация за тази промяна. Може да използваме информацията за първата клетка след променената. Отговорът би бил същия с изключение на един специален случай, ако от първата клетка след променената има път до променената. Тогава ще зациклим.

Задачата се свежда до това да намерим дали от една клетка имаме път до друга и ако да, то колко е разстоянието между двете клетки. Това ще търсим във всяка от следващите подзадачи.

**Подзадача 2.** За тази задача не са нужни всички предишни наблюдения понеже изход има само ако от началната клетка минаваме през променената и от променената се излиза директно извън лабиринта. Но пак остава въпроса дали от началната се стига до променената и какво е разстоянието между двете. За да видим дали има път е лесно, може в началото да си отбележим клетките, които са част от един цикъл. Например може да използваме алгоритъма за свързани компоненти и да си номерираме всяка клетка в кой свързан компонент участва. Сега ако имаме две клетки в един цикъл идва по-сложното – да намерим разстоянието между тях. За всеки цикъл може да изберем една отправна клетка (произволна клетка от цикъла) и да намерим разстоянието от всяка клетка в цикъла до отправната клетка. За разстоянието между две клетки A и B ще имаме два случая - ако от A първо се стига до B, после до отправната клетка и ако от А минаваме през отправната клека, след това стигаме до B. За сметките допълнително ще ни трябва и дължината на цикъла.

**Подзадача 3.** Това, че от всяка клетка се стига до изход, означава че всички клетки, които водят към един и същи изход образуват дърво, погледнато наобратно. Да проверим дали от клетка A се стига до клетка B за нашето дърво означава А да се намира в поддървото на B. Може да се използва алгоритъм за най-близък общ предшественик. Намирането на разстоянието се свежда до намиране на дълбочината на върховете в дървото.

**Подзадача 4.** Разликата с подзадача 3 е, че не винаги имаме изход в началото. Ако в даден цикъл отрежем пътя между две съседни клетки, ще получим дърво подобно на подзадача 3 и корена ще е една до отрязания път. Така може да направим проверката дали има път в дървото. Заради отрязания път може да стигнем до отреза, а не до целта. Това може да се случи само ако целта е част от цикъла. Но в този случай винаги ще имаме път. Така имаме три случая за проверката на път от клетка A до B:

* Ако A и B са части от различни изходни или отрязани дървета
* Ако A е в под-дървото на B
* Ако A и B водят към един и същ корен на отрязано дърво и B е част от цикъла, преди отрязването

По-сложната част идва от търсенето на разстоянието от A до B. Когато A е в под-дървото на B, то от A се стига до B после до отреза и ще гледаме дълбочината в дървото. В последния случай от А минаваме през отреза после стигаме до B. Трябва освен дълбочината на дървото за всеки връх да пазим на какво разстояние влиза в цикъла.

**Подзадача 5.** Единствената разлика е начина на проверка дали един връх е в под-дървото на друг. Това може да се реши линейно като се направи обхождане в дълбочина и се запишат часовете на влизане и излизане във всеки връх. Ако А e в под-дървото на B, то обхождането по ред влиза в B, влиза в A, излиза от A, излиза от B.

**Подзадача 6.** Верни подзадачи 4 и 5 трябва да водят до пълно решение на цялата задача.

Петър Петров