ЗАДАЧА ЗА ТЪРГОВСКИЯ ПЪТНИК

Даден е пълен неориентиран граф G с положителни тегла на ребрата. Да се намери най-късият хамилтонов цикъл в G. С други думи, търси се най-късият затворен маршрут, който минава веднъж през всеки връх на графа. Теглото на всяко ребро се тълкува като разстояние между краищата му, тоест дължината на маршрут е равна на сбора от теглата на ребрата от маршрута.

Изискването за положителни тегла не е съществено: като прибавим достатъчно голямо число R към всички тегла, те ще станат положителни числа, а пък дължините на всички хамилтонови цикли ще се увеличат с едно и също число, а именно nR, където n е броят на върховете на G. Ето защо най-късият хамилтонов цикъл ще остане най-къс.

От друга страна, в практически задачи теглата обикновено са положителни, защото означават действителни разстояния между някакви обекти. Названието на задачата произлиза от тълкуването, според което върховете на графа означават населени места, свързани с шосета (ребрата на графа). Търговски пътник от едно населено място трябва да обиколи по веднъж останалите населени места и да се върне там, откъдето е тръгнал. Естествено, търговският пътник иска да намери възможно най-къс маршрут.

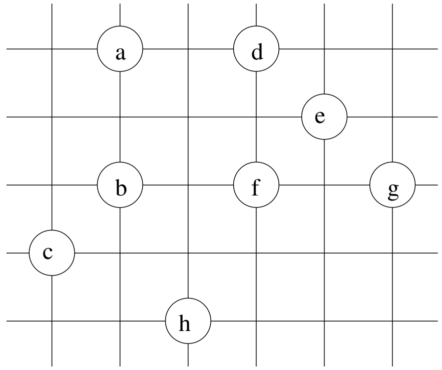
Тази задача е оптимизационна. Съответната задача за разпознаване ("В даден пълен тегловен неориентиран граф G съществува ли хамилтонов цикъл с дължина, ненадхвърляща дадено число L ?") е NP-пълна.

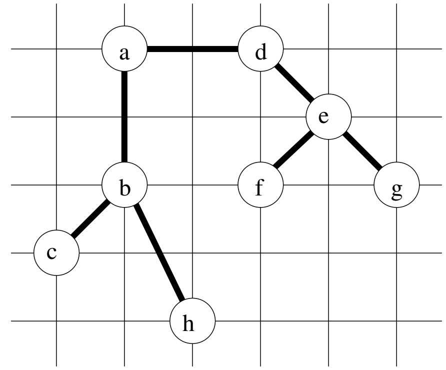
След като задачата за търговския пътник е NP-пълна, то тя вероятно не притежава алгоритъм с полиномиална времева сложност (при най-лоши входни данни). Възниква необходимост от търсене на приблизително решение.

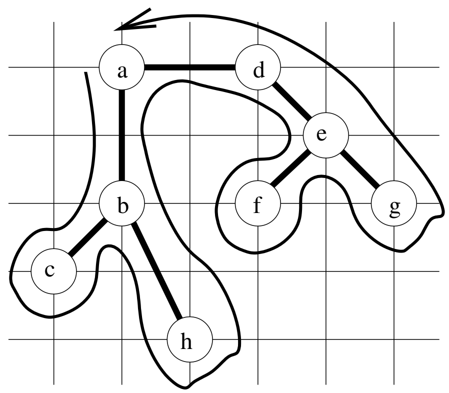
Нека върховете на графа са точки в евклидовата равнина и дължината на всяко ребро е равна на евклидовото разстояние между краищата му (т.е. ребрата на графа са просто отсечки). Тогава дължините на ребрата удовлетворяват неравенството на триъгълника, което позволява съставянето на апроксимиращ алгоритъм с полиномиална времева сложност при най-лоши входни данни:

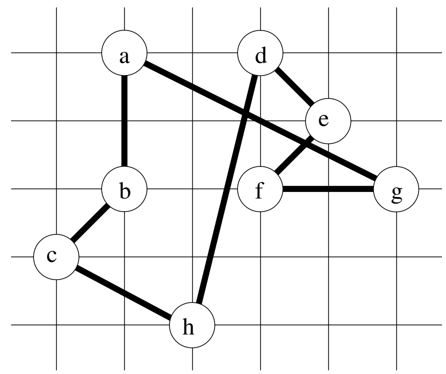
1) Построяваме минимално покриващо дърво на графа (например с алгоритъма на Прим——Ярник).
2) Извършваме обхождане от тип preorder на дървото. (Нагледно: рисуваме контур около него, при което повтаряме върховете, където е необходимо; след туй премахваме повторенията.) Полученият цикличен списък от върхове съдържа всеки от върховете на графа по един път, затова може да се тълкува като хамилтонов цикъл. Този цикъл е резултатът на алгоритъма.

Алгоритъмът е полиномиален, защото такава е всяка от двете стъпки. Стъпка № 1 има времева сложност $m + n \log n$, ако алгоритъмът на Прим——Ярник се реализира чрез пирамидата на Фибоначи. Стъпка № 2 има времева сложност n. Ето защо времевата сложност на целия алгоритъм се определя от сложността на първата стъпка, тоест общото време на алгоритъма е от порядък $m + n \log n$.

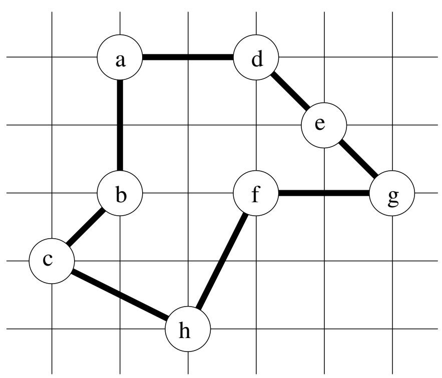








Резултат: 19,1.



Верен отговор: 14,7.

Нека H е дължината на хамилтоновия цикъл, върнат от алгоритъма, а h е най-малката дължина на хамилтонов цикъл в дадения граф. Нека T да е теглото (т.е. сборът от дължините на ребрата) на построеното минимално покриващо дърво.

Поради минималността на h важи неравенството $h \leq H$.

От най-къс хамилтонов цикъл (тоест такъв с дължина h) да изтрием произволно избрано ребро, чиято дължина да означим със $d > 0$ (или $d \geq 0$, ако допуснем и нулеви тегла). След изтриването остава хамилтонов път с дължина (тегло) $h - d \leq h$. Ала всеки хамилтонов път е покриващо дърво, следователно $T \leq h - d$ поради минималността на T. От последните две неравенства се получава ново неравенство: $T \leq h$.

Докато рисуваме контура около дървото, обхождаме всяко негово ребро точно два пъти, затуй дължината на цикъла, получен в този миг, е равна на $2T$. При премахване на повторените върхове обиколните пътища се заменят с преки; поради неравенството на триъгълника дължината на цикъла намалява или остава същата при всяко изтриване. Ето защо последната му дължина не е по-голяма от първоначалната, тоест $H \leq 2T \leq 2h$.

И така, предложеният алгоритъм намира хамилтонов цикъл с дължина H, за която $h \leq H \leq 2h$. Тоест дължината на намерения хамилтонов цикъл може да надвишава минимума най-много два пъти.

Разсъжденията по-горе зависят съществено от неравенството на триъгълника. Ако то не важи, оценката на точността на приближението губи силата си.

В общия случай (когато неравенството на триъгълника може да не е изпълнено) се оказва, че не съществува апроксимиращ алгоритъм с фиксирана относителна точност, както и да е избрана тя: ако $P \neq NP$, то колкото и голямо число $r > 1$ да изберем, няма алгоритъм, който да може да реши задачата за търговския пътник за полиномиално време при всякакви входни данни, като гарантира, че дължината на намерения хамилтонов цикъл надхвърля минималната не повече от r пъти.

Доказателството се извършва с полиномиална редукция от NP-трудната задача за разпознаване на хамилтонови графи. Да допуснем противното — че за подходящо $r > 1$ съществува полиномиален алгоритъм ALG с фиксирана относителна точност r. Разглеждаме следната полиномиална редукция:

ХАМИЛТОНОВ ГРАФ (G: неориентиран нетегловен граф с n върха и m ребра)

1) На всички m ребра на G даваме тегло 1.
2) Добавяме липсващите $n(n-1)/2 - m$ ребра.
3) На добавените ребра даваме тегло $rn + 1$.
4) Върху така получения (тегловен) граф G извикваме алгоритъма ALG,
   тоест $H \leftarrow ALG(G)$, където H е число (дължината на хамилтоновия цикъл, намерен от ALG).
5) Ако $H < (r+1)n$,
   връщаме "да" (т.е. G е хамилтонов граф),
   иначе връщаме "не" (т.е. G не е хамилтонов граф).

Първите три стъпки се изпълняват за толкова време, колкото са ребрата на пълния граф, т.е. $n(n-1)/2$, а това е полином. Алгоритъмът ALG по допускане се изпълнява за полиномиално време. Последната стъпка (събиране, умножение и сравнение) изразходва 3 единици време (също полином). Времевата сложност на алгоритъма е сборът от времената на стъпките, тоест тя е полиномиална.

Коректност на алгоритъма: Можем да приемем, че задачата за търговския пътник се разглежда само при $n > 2$ (обратният случай е безинтересен). При $n > 2$ допълненият, вече тегловен граф G съдържа поне един хамилтонов цикъл; нека h е най-малката дължина на такъв цикъл.

Ако в първоначалния (нетегловния) граф G има хамилтонов цикъл, допълненият граф G съдържа хамилтонов цикъл, съставен само от стари ребра. Те имат тегло 1, т.е. дължината на цикъла е n. Другите ребра имат по-големи дължини ($r > 1$), т.е. $h = n$. Всеки друг хамилтонов цикъл съдържа поне едно ново ребро, затова дължината му е поне $(rn + 1) + (n - 1) = (r + 1)n$. Следователно, ако първоначалният граф не е хамилтонов, то $h \geq (r + 1)n$. Разглеждаме двата случая поотделно.

Ако първоначалният (нетегловният) граф G не е хамилтонов, то за получения тегловен граф G $h \geq (r + 1)n$. Понеже $H \geq h$, то ALG връща някакво число $H \geq (r + 1)n$ и на стъпка № 5 решаваме, че първоначалният граф G не е хамилтонов, което е точно така.

Ако първоначалният граф е хамилтонов, то $h = n$. По допускане ALG е апроксимиращ алгоритъм с фиксирана относителна точност r, тоест ALG връща H, за което $h \leq H \leq rh$. Но тъй като $h = n$, то $n \leq H \leq rn < (r + 1)n$ и на стъпка № 5 правилно решаваме, че оригиналният граф е хамилтонов.

Съставихме полиномиален алгоритъм за NP-пълна задача, значи $P = NP$, което е противоречие.

Доказателството, следователно и твърдението за липса на полиномиален апроксимиращ алгоритъм остават в сила и тогава, когато r, вместо да е фиксирано, е произволна функция на n, изчислима за полиномиално време.