АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
МАТРИЦА

Решението на задачата се основава на следното съображение: ако могат да бъдат намерени такива две редици от цели неотрицателни числа ($m_1, m_2, \ldots\ldots\ldots\ldots, m_N$) и ($w_1, w_2, \ldots\ldots\ldots\ldots, w_N$), които удовлетворяват условието (1) от условието на задачата и за които съществува такава пермутация ($j_1, j_2, \ldots\ldots\ldots j_N$) на числата (1, 2,………N), че

$$m_i + w_{j_i} = c_{ij_i} \text{ за всяко } 1 \le i \le \mathrm{N},$$

то сумата от числата в тези две редици ще бъде търсената минимална сума. Наистина, сумата $\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} w_i$ може да бъде записана като $(m_1 + w_{j_1}) + (m_2 + w_{j_2}) + ...(m_N + w_{j_N})$ и от този запис се вижда, че, ако съществува възможност за намаляване на сумата, това би означавало, че трябва да намалее поне едно от събираемите $m_i + w_{j_i}$, което би довело до нарушаване на условие (1).

Това съображение лежи в основата и на двата алгоритъма за решаване на задачата, изложени по-долу.

Алгоритъм, базиран на системи от различни представители.

1. Намират се начални редици ($m_1, m_2, \ldots\ldots\ldots\ldots, m_N$) и ($w_1, w_2, \ldots\ldots\ldots\ldots, w_N$), които удовлетворяват условието (1). Това става като w_i приеме стойност 0 за всяко i, а m_i приеме стойност, равна на максимума на c_{ij} , когато j се изменя от 1 до N. След това начално определяне на редиците се изпънява следният цикъл:
2. За всяко 1≤i≤N се формира множество S_i от индекси j, за които $m_i + w_j = c_{ij}$ (тези множества няма да бъдат празни – при първата итерация това се гарантира от т.1, а при следващите от т.4).
3. Търси се **система от различни представители(срп)** за множествата $S_1, S_2, \ldots\ldots S_N$. Ако такава система съществува, то тогава е намерена пермутацията от индекси j, за която стана въпрос по-нагоре и задачата е решена. Условието да съществува **срп** е всяка съвкупност от k на брой множества S да съдържа общо поне k на брой различни индекса. Ако не съществува **срп**, то това значи, че съществува множество K от k на брой редове (индекси i), за които общият брой индекси j в множествата им S е t<k. Нека множеството от тези индекси j означим с J. Тогава да извършим следните преобразувания в стойностите на m и w: намаляваме с 1 стойностите на m_i, за тези индекси i, които принадлежат на множеството K и увеличаваме с 1 стойностите на w_j за онези индекси j, които принадлежат на множеството J. Стойностите на останалите m и w остават непроменени. Лесно се проверява, че условие (1) продължава да се изпълнява, а общата сума е намаляла с (k-t).
4. Тъй като в резултат на изпълнението на т.3. могат да се получат редове i, в които да няма нито един индекс j, такъв че се изпълнява условието $m_i + w_j = c_{ij}$, то се изпълнява допълнително намаляване на стойностите на m_i, така че да се запази изпълнението на условие (1) и в същото време за всяко i да съществува поне един индекс j, такъв, че $m_i + w_j = c_{ij}$.
5. Преминава се към точка 2.

Тъй като на всяка итерация общата сума се намалява, то ще се стигне до момент, в който ще бъде намерена **срп** и задачата ще бъде решена.

За намиране на **срп** могат да бъдат използвани различни алгоритми, които трябва да са известни на състезателите.

Решението се оптимизира по бързодействие като изменението на стойностите на m и w се извършва не с 1, а с минимума на разликата $(m_i + w_j) - c_{ij}$ по всички индекси i и онези индекси j, които не принадлежат на множествата S_i.

Това решение представлява по-скоро теоретичен интерес от гледна точка на прилагането на **срп** за решаването на оптимизационна задача. Изчислителната сложност на алгоритъма трудно може да бъде определена, тъй като не е ясно колко пъти ще се наложи да бъде търсена система от различни представители (аз поне не можах да намеря задоволителна оценка).

Алгоритъм със сложност $O(n^3)$.

Идеята за този алгоритъм е заимствана от унгарския алгоритъм за решаване на задачата за назначенията.

1. Намират се начални редици (m_1, m_2,……………,m_N) и (w_1, w_2,……………,w_N), които удовлетворяват условието (1). Нека в този случай това става като m_i приеме стойност 0 за всяко i, а w_j приеме стойност, равна на максимума на c_{ij} , когато i се изменя от 1 до N. Т.е. водещи в този алгоритъм ще бъдат стълбовете на матрицата.
2. Първа, подготвителна стъпка. Намираме множество от такива двойки индекси (i,j), че $m_i + w_j = c_{ij}$ за всяка от тях и никои две двойки не са от един ред или от един стълб (такова множество от двойки ще наричаме *независимо*). Това става по следния начин: точка 1 гарантира, че за всеки стълб има поне един ред, за който се изпълнява условието $m_i + w_j = c_{ij}$. От първия стълб вземаме една такава двойка. След това преминаваме към втория стълб и, ако там има двойка удовлетворяваща условието $m_i + w_j = c_{ij}$ и не съвпадаща с реда на двойката от първия ред, то я добавяме към независимото множество. По същия начин преминаваме през останалите стълбове, като всеки път търсим двойка, удовлетворяваща условието $m_i + w_j = c_{ij}$ и такава, че нейният ред да не се появявал в двойките от построеното вече независимо множество. Забележете, че никаква максималност на построяваното множество не се търси. В матрицата маркираме с „*” елементите, които съответстват на двойките от независимото множество, а с „+” стълбовете, в които са тези двойки. Ще смятаме, че един елемент от матрицата е **маркиран**, ако той самият е маркиран или лежи в маркиран стълб или ред (защото по-нататък ще се появят и маркирани редове).

   Ако броят на двойките в независимото множество е равен на N, то е намерено решение на задачата.

   Ако броят на двойките в независимото множество е по-малък от N, то се преминава към изпълнението на цикъл от итерации, като целта на всяка итерация е да се увеличи с 1 броят на двойките в независимото множество (при този процес някои от двойките, получени след предишната итерация могат да се разпаднат и да се получат нови).
3. В рамките на една итерация се прави следното:
   a. Сред немаркираните елементи на матрицата търсим такъв (с индекси i и j), че да е изпълнено условието $m_i + w_j = c_{ij}$. Възможни са два случая – такъв елемент съществува или не съществува.
      i. Ако такъв елемент съществува, то го маркираме с „!”(потенциален кандидат неговите индекси да влязат в независимото множество). В този

случай са възможни два подслучая – в реда, в който той се намира да има елемент с индекси от независимото множество (маркиран с „*”) или да няма. Ако има, то махаме маркировката на стълба, в който е елементът, маркиран с „*” и маркираме с „+” реда, в който се намират двата елемента(забележете, че не махаме маркировката на самия елемент, чийто индекси все още са в независимото множество, т.е. той продължава да бъде маркиран с „*”).

ii. Ако такъв елемент не съществува, то намираме минимума на разликата $(m_i + w_j) - c_{ij}$ за всички двойки индекси (*i,j*), които съответстват на немаркирани елементи от матрицата (това ще бъде положително число). Този минимум вадим от w_j за всички немаркирани стълбове и добавяме към всички маркирани. Лесно се съобразява, че тази операция не променя изпълнението на условието $(m_i + w_j) \geq c_{ij}$ за всяка двойка (*i,j*), но в резултат на нея се появяват немаркирани елементи (поне един), за които е изпълнено $m_i + w_j = c_{ij}$. Това дава възможност да се върнем към изпълнението на стъпка а. като в този случай ще попаднем в случай i.

Стъпка а. се изпълнява докато, при изпълнението на случай i, не се стигне до ситуация, в която в реда на маркирания с „!” елемент няма елемент, маркиран с "*" (чийто индекси са от независимото множество). Достигането до такъв елемент (не забравяйте, че по време на неговото търсене сме изменяли сумите на m_i и w_i) дава отправната точка, да се преправи независимото множество, като броят на двойките в него се увеличи с 1. За целта се изпълнява стъпка b.

b. Тръгвайки от последния, маркиран с „!” елемент (в неговия ред няма маркиран с „*” елемент) се прави следното – в негови стълб се търси елемент, маркиран с „*”. Ако такъв елемент бъде намерен, то в неговия ред се търси елемент, маркиран с „!”, след това, ако такъв елемент бъде намерен, в неговия стълб се търси елемент, маткиран с „*” и т.н. Процедурата продължава, докато в стълба на елемент, маркиран с „!” не се намери елемент, маркиран с „*” (обяснете защо тази верига от елементи, започваща с елемент, маркиран с „!” задължително трябва да завърши отново с елемент, маркиран с „!”). По този начин във веригата елементите, маркирани с „!” са с 1 повече от елементите, маркирани с „*”. Тогава размаркираме ***само елементите от получената верига***, маркирани с „*” (вадим техните двойки индекси от независимото множество) и маркираме с „*” елементите ***от веригата***, маркирани с „!”(вкарваме техните двойки индекси в независимото множество, увеличавайки по този начин броя на двойките в него с 1).

Ако броят на двойките в независимото множество е станал N, то задачата е решена, иначе премахваме всички маркировки „!” от елементите, които не са влезли във веригата, премахваме всички стари маркировки „+” от редовете и стълбовете и маркираме с „+” стълбовете, в които има елементи, маркирани с „*”, след което преминаваме към изпълнението на следващата итерация.

Автор: Руско Шиков