CODE FESTIVAL 2016 Exhibition A 問題 解説

snuke

まず、頂点 1 からの距離 A_i が満たしているべき条件を考える。

- $A_1 = 0$
- すべての頂点 $v(v \neq 1)$ について $A_v \neq 0$
- すべての頂点 $v(v \neq 1)$ について $A_v - 1 = A_u$ を満たす辺 (u, v) が存在する
- すべての辺 (u, v) について $|A_u - A_v| \leq 1$

頂点 v までの最短距離が A_v 以下であることが 3 番目の条件から、A_v 以上であることが 4 番目の条件から言え、これらの条件さえ満たしていれば良いことが確認できる。

(A_i, B_i) を二次元平面上にプロットする。頂点 i からは、以下の 2 種類の辺を張らなければならない。それぞれの辺を「A 用の辺」「B 用の辺」と呼ぶことにする。

- $(A_i - 1, B_i - 1)$ または $(A_i - 1, B_i)$ または $(A_i - 1, B_i + 1)$ の頂点に 1 本
- $(A_i - 1, B_i - 1)$ または $(A_i, B_i - 1)$ または $(A_i + 1, B_i - 1)$ の頂点に 1 本

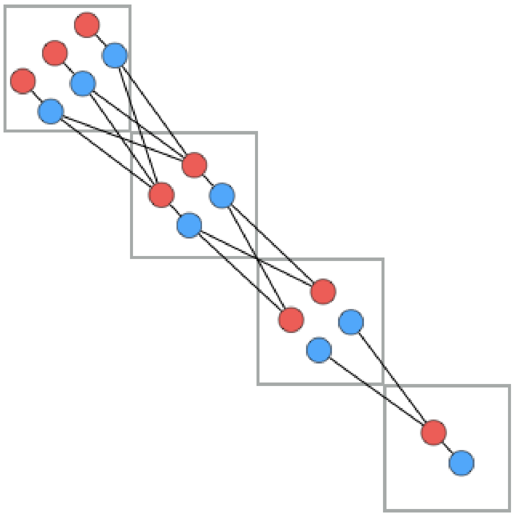
例外として、頂点 1 は A 用の辺を張る必要がなく、頂点 2 は B 用の辺を張る必要がない。A 用の辺または B 用の辺を張ることができない場合、すなわち該当する座標に頂点がない場合は、距離の条件を満たすグラフは存在しない。逆に、すべての頂点について A 用の辺または B 用の辺を張ることができる場合は、それらの $2N - 2$ 本の辺をすべて繋ぐことにより距離の条件を満たすグラフを構成することができる。

少なくとも $2N - 2$ 本の辺がグラフを構成できることは分かったが、この問題では辺の本数の最小値を求めなければいけないため、辺の本数を減らすことを考えなければならない。辺の本数を減らすためには 2 つの用途を兼ねた辺を作るしかなく、そのような辺の本数を最大にしたい。2 つの用途を兼ねた辺を作る方法は以下の 2 通り考えられる。それぞれ「タイプ X」「タイプ Y」と呼ぶことにする。

- 頂点 i の A 用と B 用を兼ねた辺 : $(A_i - 1, B_i - 1)$ に頂点が存在するときのみ可能
- 頂点 i の A 用と頂点 j の B 用を兼ねた辺 : $A_i - 1 = A_j$ かつ $B_i = B_j - 1$ を満

たすときのみ可能

「A 用の辺」「B 用の辺」を頂点だと考え、2 つの用途を兼ねる辺が張れるような頂点の間に辺を張ったグラフを考えると、このグラフでの最大マッチングを求められれば良いことになる。$A_i + B_i$ の値が異なる頂点の用途を兼ねられる場合が無いことに注目すると、$A_i + B_i$ の値で頂点を分類し、それぞれを独立に考えることができることが分かる。下図は上記のようなグラフから $A_i + B_i$ が等しい頂点の部分だけを抜き出して図示したものである。赤い頂点は「A 用の辺」、青い頂点は「B 用の辺」を表しており、灰色の枠内に張られている辺はタイプ X の辺、灰色の枠をまたいで張られている辺はタイプ Y の辺を表している。タイプ X の辺がある場所とない場所が存在することに注意したい。



このようなグラフの最大マッチングは、「左上の辺から順に可能な限りマッチングさせていく」という貪欲法で求めることができる。答えは $2N - 2 - ($最大マッチング$)$ となる。

$A_i = 0$ や $B_i = 0$ のケースには注意しなければならないだろう。たとえば、$A_3 = 0$ の場合などは不可能だと判定しなければならない。

CODE FESTIVAL 2016 Exhibition B 問題 解説

chokudai

まず、同じ硬貨を 10 枚以上持つ必要は全く存在しない。例えば、 1 ミョン硬貨を 10 枚持った時に払うことが可能な金額の集合は、1 ミョン硬貨を 9 枚と、10 ミョン硬貨を 1 枚持った時に支払い可能な金額の集合の部分集合になっているからである。そこで、各硬貨について、9 枚以下しか使わない、という前提を持って良い。

ここで、各硬貨についてそれぞれ何枚必要かについて順番に考えることとする。注目している硬貨を i 番目の硬貨とし、この金額を 10^i 円とする。

仮に $i = 3$ のケースについて考える。 1000 ミョン硬貨を 9 枚必要にするには、10000 で割った余りが 9000 から 9999 になれば良い。言い換えれば、あと 9000 から 9999 の支払いが出来る組み合わせが存在すれば良い。ここで、2000 ミョンの品物があったとする。この品物を選んだ場合、残りの支払い金額は 7000 から 7999 となり、使わなかった場合は、残りの支払い金額は 9000 から 9999 である。このように、9 枚必要となる範囲を持ち、これを更新していく DP が出来れば、9 枚必要になるか否かは求めることが可能となる。この区間が 0 に掛かるのであれば、9 枚の硬貨が必要であるといえる。

この DP を考えた時に、保持するべき区間の長さは、必ず全区間の 1/10 以上の長さとなる。ここで、全ての区間が重なりを持たないとき、区間の個数は 9 個以下となる。区間の個数が十分に少ない定数個なので、区間を更新する DP を行う計算量は O(N) で済む。

8 枚以下についても同様に求めることが可能である。1 回の DP でいくつ必要かを求める方法も存在するが、ここでは割愛する。

ここまで出来れば、各硬貨について同じ処理を行うことで、必要枚数を求めることが出来、その枚数を足したものが、求める答えとなる。

CODE FESTIVAL 2016 Exhibition Problem A Editorial

snuke

First, we will consider the conditions that A_i, the distances from vertex 1, should satisfy:

- $A_1 = 0$
- For any vertex $v (v \neq 1)$, $A_v \neq 0$
- For any vertex $v (v \neq 1)$, there exists an edge (u, v) such that $A_v - 1 = A_u$
- For any edge (u, v), $|A_u - A_v| \leq 1$

It can be shown from the third condition that the shortest distance from vertex 1 to vertex v is at most A_v, and it can also be shown from the fourth condition that the shortest distance is at least A_v, thus these four conditions are sufficient.

We we will plot the points (A_i, B_i) for all vertices i on a two-dimensional plane. From each vertex i, the following two edges must be spanned:

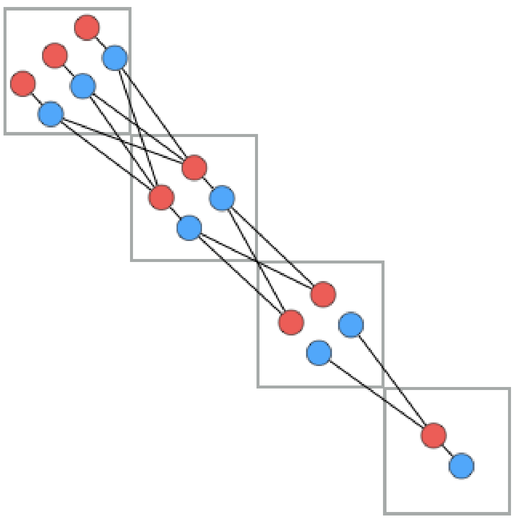
- An edge to one of the vertices at the following points: (A_i-1, B_i-1), (A_i-1, B_i) and (A_i-1, B_i+1)
- An edge to one of the vertices at the following points: (A_i-1, B_i-1), (A_i, B_i-1) and (A_i+1, B_i-1)

We will refer to the former as an A-edge, and the latter as a B-edge. The exceptions is that no A-edge needs to be spanned from vertex 1, and no B-edge needs to be spanned from vertex 2. If an A-edge or B-edge cannot be spanned, that is, there is no vertex at the corresponding coordinates, there exists no graph satisfying the conditions. Conversely, if all the A-edges and B-edges can be spanned, a graph satisfying the conditions can be constructed by spanning all those $2N-2$ edges.

We have found that at most $2N-2$ edges are necessary, but we must find the minimum possible number of edges. Let us consider how to reduce the number of edges. The only possible way to reduce the number is to use an edge as both A-edge and B-edge. There are two possible situations where an edge is used as both types of edges:

- An edge used as both A-edge and B-edge of vertex i: possible only when there is a vertex at $(A_i - 1, B_i - 1)$
- An edge used as A-edge of vertex i and B-edge of vertex j: possible only when $A_i - 1 = A_j$ and $B_i = B_j - 1$

We will refer to the former as an X-edge, and the latter as a Y-edge. When we consider a graph with a vertex for each A-edge and B-edge, and two vertices are connected by an edge when an X-edge or Y-edge can be used as the corresponding A-edge and B-edge, what we want to find is the maximum matching in this graph. We will focus on the fact that no X-edge or Y-edge can be used as both A-edge and B-edge whose values of $A_i + B_i$ are different, and classify the vertices in this graph according to the value of $A_i + B_i$, then each category can be considered independently. The figure below shows part of a graph described above corresponding to one such category. An red vertex represents an A-edge, a blue vertex represents a B-edge, an edge spanned within a gray frame represents an X-edge, and an edge spanned across gray frames represents a Y-edge. Note that there are frames in which X-edges are spanned, and there are also frames in which X-edges are not spanned.



A maximum matching in such a graph can be found by a greedy strategy to use as much edges as possible from the upper left to the lower right. The answer will be $2N - 2 -$ (size of the maximum matching).

Pay attention to the case where $A_i = 0$ or $B_i = 0$. For instance, if $A_3 = 0$, there exists no graph satisfying the conditions.

CODE FESTIVAL 2016 Exhibition Problem B Editorial

chokudai

First, observe that there is no need to have ten or more coins in the same denomination. For example, the set of the amounts of money that can be paid with ten 1-Myon coins is a subset of the set of the amounts of money that can be paid with nine 1-Myon coins and one 10-Myon coins. Thus, we can assume that we bring at most nine coins in the same denomination.

Now, we will separately consider the number of coins we need in each denomination. Suppose we are now dealing with 10^i-Myon coins.

Consider the case where $i = 3$. When we need nine 1000-Myon coins? We need nine of them when the total price is between 9000 and 9999, inclusive, modulo 10000. That is, when there exists a set of items sold at the store for the total price of between 9000 and 9999, modulo 10000. We will now decide whether we take each item. Suppose we decided to take a item with the price of 2000 Myon. Then, we will need nine 1000-Myon coins if the total price of items we take afterwards is between 7000 and 7999, modulo 10000. If we decided not to take the item instead, we will need nine 1000-Myon coins if the total price of items we take afterwards is between 9000 and 9999, modulo 10000. In this manner, we can determine whether we need nine 1000-Myon coins by taking a DP approach, maintaining the ranges corresponding to when we need nine 1000-Myon coins. If the ranges eventually cover 0, we can say we need nine 1000-Myon coins.

During the DP approach, each of the ranges that has to be maintained has a length of at least one-tenth of the whole segment. Thus, the number of non-overlapping ranges maintained is at most nine, which is a reasonably small constant and the DP approach maintaining ranges takes only $\mathrm{O}(N)$ time.

Similarly, we can determine whether we need eight 1000-Myon coins and so forth, to find the necessary number of 1000-Myon coins. (It is possible to determine the necessary number by executing the DP only once, but it is not required.)

Now, find the necessary number of coins in every denomination, and sum up the results to obtain the answer.