MAC0345 - Desafios 2
Editorial Lista 14

Enrique Junchaya

July 12, 2023

A. Fat hobbits.
Tópicos: Dilworth

Podemos modelar os pesos dos hobbits com um *poset* (*partially ordered set*). O problema pede para achar o conjunto maior de hobbits cujos pesos não da para comparar. Em outras palavras, queremos achar uma anticadeia máxima do *poset*.

Como vimos em aula, o teorema de Dilworth nos diz que a cardinalidade de uma anticadeia máxima é igual à cardinalidade de uma menor partição em cadeias do *poset*. Modelando o *poset* como uma DAG, uma mínima partição em cadeias é um *minimum path cover* da DAG. Em aula vimos como resolver esse último problema usando o *maximum matching* de um grafo bipartido construído a partir da DAG.

Para obter a anticadeia máxima em si, veja o complemento do *minimum vertex cover* do grafo bipartido construído. É possível provar que os elementos do *poset* associados aos vértices do complemento descrito formam uma anticadeia cuja cardinalidade é maior ou igual n - m, onde n é o número de elementos no *poset* e m é a cardinalidade do emparelhamento máximo do grafo bipartido construído. De novo, usando o teorema de Dilworth, temos que tal anticadeia é de fato máxima.

B. Taxi.
Tópicos: bipartite matching

O problema consiste em emparelhar taxistas com passageiros. Se modelassemos os taxistas e os passageiros como vértices de um grafo e colocamos arestas entre um taxista e um passageiro se ele consegue ir à posição do passageiro no máximo em c segundos, claramente teríamos um grafo bipartido (um lado da bipartição representaria aos taxistas e o outro aos passageiros). Desta forma, o problema se reduz a achar um emparelhamento máximo num grafo bipartido.

C. Array and operations.
Tópicos: bipartite matching, teoria dos números

Veja que se temos dois números que estão num *good pair* e queremos dividir eles entre algum inteiro v que divida ambos números, nunca faz sentido que v não seja primo porque queremos maximizar o número de vezes em que fazemos essas operações. Veja que se v não fosse primo, então teríamos $v = pq$ com p e q inteiros. Assim, poderíamos ter feito duas operações dividindo por p e por q em vez de apenas uma. É natural, portanto, pensar na decomposição em primos dos números dados.

Se modelamos o problema com um grafo em que cada vértice representa um fator primo dos números da entrada e que temos arestas entre vértices i e j se i e j são fatores primos IGUAIS de *good pairs*. Por exemplo se os números 12 e 15 são *good pairs*, teríamos uma aresta entre os vértices representando o fator 3 de ambos números. Note que podemos representar um conjunto de divisões feita como um emparelhamento desse grafo: se a aresta ij está no matching, então i e j representam um mesmo número primo que foi usado para dividir os números dos que são fatores. É importante que seja um matching e não qualquer coleção de arestas porque não podemos usar o mesmo fator em mais de uma divisão. Desta forma, reduzimos o problema de achar uma forma de maximizar o número de operações feitas ao emparelhamento máximo do grafo construído.

Finalmente, veja que a condição de que $i + j$ seja ímpar, onde (i, j) é um *good pair* faz que o grafo construído seja bipartido. Ter $i+j$ ímpar faz que exatamente uma dessas parcelas seja par e a outra ímpar. Assim, temos uma bipartição do grafo: uma partição são os números com índice ímpar e a outra os que tem índice par.

D. Girls and boys.

Tópicos: bipartite matching, teoria de grafos

Já que (na questão) existem apenas relacionamentos entre meninos e meninas, podemos modelar o problema com um grafo bipartido: um lado da partição são os meninos, o outro lado as meninas e temos uma aresta num menino e uma menina se eles já se envolveram romanticamente. Desta forma, o problema se reduz a achar o máximo conjunto independente do grafo.

Como vimos em aula, num grafo bipartido, vale que $\nu(G)+\alpha(G) = |V(G)|$, onde $\nu(G)$ é igual ao tamanho de um emparelhamento máximo de G, $\alpha(G)$ é igual ao tamanho de um máximo conjunto independente de G e $V(G)$ é o conjunto de vértices do grafo. Desta forma, para calcular o tamanho de um máximo conjunto independente, basta saber o tamanho de um emparelhamento máximo do grafo.

E. Arranging flowers.

Tópicos: bipartite matching, teoria de grafos, guloso

O retângulo dado no problema é um retângulo latino. Primeiro, vamos mostrar que é sempre possível adicionar filas ao retângulo latino até completar um quadrado latino.

Vamos, claro, a modelar o problema como um grafo. Pense nas colunas do retângulo e nos tipos de flores (números que estão nas células do retângulo) como vértices. Cada aresta do grafo liga um vértice que representa uma coluna j com um vértice que representa um tipo flor f se e somente se f não aparece na coluna j do retângulo (ou seja, que ainda podemos adicionar f na coluna j em alguma fila posterior). É claro que o grafo descrito é bipartido e k-regular (todos os vértices tem grau k), onde $k = n - m$. Note que um emparelhamento perfeito do grafo representa uma forma de preencher uma nova fila com a que podemos estender o retângulo. Vamos mostrar que, se $m < n$, esse grafo sempre tem um emparelhamento perfeito (sempre é possível adicionar uma fila a um retângulo latino) e, então, por indução, concluir que sempre podemos completar o quadrado.

Sejam A e B as bipartições do grafo. Vamos provar que para todo conjunto de vértices $X \subseteq A$, temos que $|N(X)| \geq |X|$, onde $N(X)$ é o conjunto de vizinhos dos vértices de X. Suponha que isso não é verdade e que existe então $X \subseteq A$ tal que $|X| < |N(X)|$. Temos $k|X|$ arestas com extremos em X, o que implica que as arestas com extremos em $N(X)$ são pelo menos $k|X| > k|N(X)|$. Contudo, isso é uma contradição porque as arestas com extremos em $N(X)$ são exatamente $k|N(X)|$. Finalmente, pelo Teorema de Hall, temos que existe um emparelhamento perfeito no grafo. Portanto, sempre é possível completar o quadrado latino adicionando uma fila de cada vez, a qual obtemos computando um emparelhamento perfeito do grafo.

Assim, para resolver a questão basta adicionar de cada vez, dentre todas as filas possíveis, a menor lexicográfica. Falta então mostrar como pegar o emparelhamento perfeito menor lexicográfico. Sabemos que sempre existe emparelhamento perfeito no grafo, então pegue um emparelhamento perfeito arbitrário M. Sabemos, também, que entre todos os emparelhamentos perfeitos do grafo, tem que existir um que seja o menor lexicográfico. Logo, pegue tal emparelhamento perfeito menor lexicográfico N. Analise a diferença simétrica dos emparelhamentos, $M \triangle N$. Tal diferença simétrica é um conjunto de ciclos MN-alternantes. Com isso, podemos pensar num método para ir de M a N trocando o emparelhamento de ciclos M-alternantes.

Itere pelos vértices que representam as colunas de esquerda a direita. Devemos minimizar o emparelhamento de cada um desses vértices nessa ordem. Uma vez que minimizemos o emparelhamento numa coluna, não devemos mexer mais nela quando formos minimizar os emparelhamentos das seguintes colunas. Vamos descrever como minimizar o emparelhamento na coluna j. Suponha que a aresta jf pertence a M.

Devemos achar um ciclo C M-alternante que contem jf, mas não contem nenhum vértice que representa as colunas menores. Seja jx a outra aresta de C que cobre j. O vértice x tem que ser o menor possível. Tem duas formas de obter isso. A primeira é tentar achar ciclos M-alternantes como o descrito começando por todas as arestas jx em ordem crescente de x. Esse método é suficiente para resolver a questão, mas tem um jeito mais eficiente. Basta achar os caminhos M-alternantes que começam em jf com uma DFS. Depois, só temos que ver o menor x tal que existe aresta entre j e x e x foi visitado na DFS (pertence a um caminho M-alternante que contem jf).

F. Dungeon of death.

Tópicos: bipartite matching, teoria de grafos

Podemos modelar a questão com um grafo. Para cada coluna e para cada fila, temos um vértice representando ela. Ligamos dois vértices se na interseção deles (no tabuleiro) tem um azulejo faltando. Queremos cobrir todos os azulejos com o número mínimo de pranchas. Pensando no grafo, queremos um conjunto mínimo de vértices tal que toda aresta é incidente a pelo menos um desses vértices. Ou seja, reduzimos a questão para o problema da cobertura mínima de vértices (*minimum vertex cover*).

Veja que o grafo é bipartido (uma partição são os vértices que representam linhas e a outra os vértices que representam colunas). Portanto, o teorema de Kőnig vale e temos que a cardinalidade da cobertura mínima de vértices é igual à cardinalidade do emparelhamento máximo do grafo.