# O problema do carteiro chinês

Carlos Eduardo Ferreira
Gabriel Fernandes de Oliveira

## 1 Problemas resolvidos

Esta seção se dedica a discutir aplicações do Problema do Carteiro Chinês em problemas de competições de programação.

### 1.1 Sereja and the Arrangement of Numbers

**Problema**

Esse problema foi publicado pelo Codeforces, para o round #215 da Div. 1, e pode ser conferido neste link.

Além do valor de $n$ também são fornecidos $m$ pares de inteiros positivos $q_i, w_i$, indicando que para se utilizar o valor $q_i$ na construção da sequência desejada há uma recompensa $w_i$. O valor da recompensa total de uma sequência é igual a soma das recompensas de seus valores distintos, ou seja, a recompensa não depende do número de vezes que cada valor aparece na sequência.

Sua tarefa é descobrir a recompensa máxima que uma sequência bonita construida usando os valores $q_i$ de tamanho $n$ pode possuir.

A entrada do programa é dada pelos valores $n, m$ e pelos $m$ pares $q_i, w_i$, garante-se que todos valores de $q_i$ são distintos.

Um exemplo de entrada para este problema é:

| | |
|---|---|
| 5 4 | Neste exemplo $n = 5$ e $m = 4$ |
| 1 4 | O primeiro valor é 1, e sua recompensa é 4 |
| 2 3 | Segue o valor 2, com recompensa 3 |
| 4 2 | Segue o valor 4, com recompensa 2 |
| 3 10 | Finalmente temos o valor 3, de recompensa 10 |

Para este exemplo, uma solução ótima é a sequência $1, 2, 3, 1, 1, 1$, que é bonita e possui recompensa total igual a 17. Não existe uma sequência bonita que possua os quatro valores do exemplo.

**Solução**

Como todo valor possui uma recompensa positiva, a solução final deverá usar a maior quantidade de valores distintos que for possível.

Começaremos discutindo um problema mais simples: Qual o tamanho da menor sequência bonita que utiliza $x$ valores distintos?

Para resolver esse problema faremos uma modelagem do problema usando um grafo completo $K_x$, onde cada vértice representa um valor distinto e cada aresta possui custo unitário.

Todo passeio do grafo $K_x$ pode ser representado como uma sequência dos valores dos vértices do grafo, as sequências definidas como bonitas são aquelas que representam um passeio que percorre todas arestas do grafo $K_x$.

Para facilitar a análise deste problema, assumiremos que pode existir em um passeio dois vértices de mesmo valor em sequência, mesmo sem haver um loop neste vértice, além disso, vamos assumir que os valores distintos são $1, 2, \ldots, x$.

Sendo assim, qualquer sequência que possua apenas valores entre 1 e $x$ pode ser representada por um passeio no grafo $K_x$.

Uma sequência $s$ bonita, de acordo com a descrição do problema, é aquela para qual exsite um inteiro $i$, para cada par de valores distintos $u$, $v$, tal que $s_i = u, s_{i+1} = v$ ou $s_i = v, s_{i+1} = u$. Interpretando a sequência $s$ como um passeio em $K_x$, podemos interpretar os valores $u, v$ como vértices do grafo construido e a existência de $i$ tal que $s_i = u, s_{i+1} = v$ ou $s_i = v, s_{i+1} = u$ como a existência da aresta $uv$ no passeio $s$. Para que a sequência $s$ seja bonita, para todo par de valores $u, v$, o passeio derivado de $s$ deverá percorrer a aresta $uv$.

Em outras palavras, para que um passeio seja bonito, ele deverá percorrer todas arestas do grafo descrito, sendo assim uma solução do problema do carteiro chinês para o grafo derivado dos valores.

Portanto, voltando a pergunta inicial, a menor sequência bonita que utiliza $x$ valores distintos, será aquela que representa uma solução ótima do problema do carteiro chinês para o grafo $K_x$, já que a solução que miniza o custo do PCC também minimizará o número de arestas percorridas, e portantom, o tamanho da sequência bonita.

Vejamos agora alguns exemplos:

Para $x = 5$, o grafo induzido é o $K_5$, como representado na figura **??**. Como todos os vértices do $K_5$ possuem grau par, a solução ótima do Problema do Carteiro Chinês para este exemplo é também um circuito euleriano, como o seguinte:

$$P = \{1, 2, 3, 4, 5, 1, 3, 5, 2, 4, 1\}$$

Como discutido anteriormente, $P$, além de solução do PCC é também uma sequência bonita que utiliza os $x$ valores definidos pelos vértices do grafo.

Sendo assim, para utilizar 5 valores distintos, é necessário ter uma sequência de tamanho no mínimo 11, valor este que é igual a $1 + |E(K_5)| = 1 + 10 = 11$.

Analisaremos agora outro exemplo, em que $x = 4$, do qual se deriva o grafo $K_4$ representado na figura **??**.

Neste exemplo, todos os vértices de $K_4$ possuem grau ímpar, por isso, a primeira parte da solução do problema do carteiro chinês se baseia em duplicar algumas arestas do grafo, tornando-o um grafo euleriano.

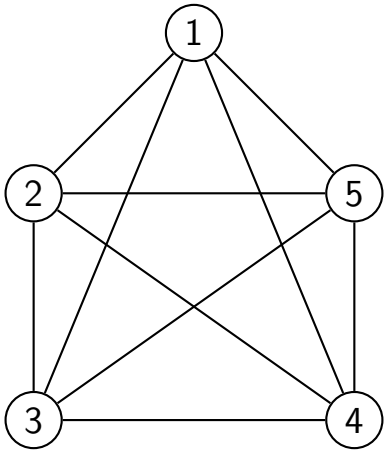


Figura 1: Exemplo com $x = 5$

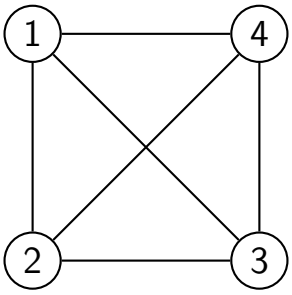


Figura 2: Exemplo com $x = 4$

Como estamos tratando de um grafo completo, com arestas de custo unitário, não é necessário criar uma condensação do grafo $K_x$, e achar um emparelhamento perfeito de custo mínimo. Basta apenas criar uma duplicação de aresta entre todo par de vértices de índices $i$ e $i + 1$ para todo índice $i$ ímpar, como representado na figura **??** para o exemplo em questão.

A partir do grafo euleriano criado, é possível encontrar o caminho euleriano $P$, que também é solução do PCC para o grafo $K_4$, e, finalmente, sequência bonita de tamanho mínimo que usa 4 valores distintos.

$$P = \{1, 2, 3, 4, 1, 3, 4, 2, 1\}$$

Sendo assim, o tamanho mínimo de uma sequência bonita que possui 4 valores distintos é 9, valor este composto da seguinte forma: $1 + |E(K_4)| = 1 + 6 = 7$ somado ao número de arestas duplicadas $\frac{4}{2} = 2$.

Podemos assim abstrair uma fórmula geral para o tamanho mínimo de uma sequência bonita $P$ que utilize $k$ valores diferentes em sua composição:

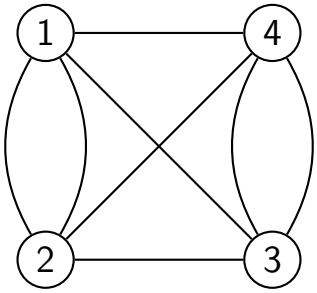


Figura 3: $K_4$ com arestas duplicadas

$$|P_k| = 1 + \frac{k(k-1)}{2} + \begin{cases} 0 & k \text{ ímpar} \\ \frac{k}{2} & k \text{ par} \end{cases}$$

Todo grafo completo com número ímpar de vértices é euleriano, enquanto os ímpares não o são. Por isso a diferença na fórmula de $|P_k|$.

Utilizando essa fórmula fechada, é possível descobrir com uma busca binária, em complexidade $\mathcal{O}(\lg n)$, qual o maior valor de $k$ tal que $|P_k| \leq n$.

Encontrado tal valor $k$, o máximo número de valores distintos que a solução pode possuir para ser bonita, basta descobrir quais valores da entrada escolher para maximizar a recompensa.

Se possuirmos na entrada uma quantidade de valores, $m$, tal que $m \leq k$, então a solução ótima será a soma das recompensas de todos valores, já que é possível encontrar uma sequência bonita que irá conter todos valores disponíveis.

Do contrário, a solução consistirá em escolher os $k$ valores de maior recompensa disponíveis. Para encontrar tal valor, basta manter uma árvore de busca binária balanceada com as $k$ maiores recompensas dos valores disponíveis, o que custa tempo $\mathcal{O}(m \lg k) = \mathcal{O}(m \lg \sqrt{n})$.

A minha solução para este problema pode ser encontrada, em C++, neste link.