Trabalho Prático 2
Índice Invertido
Algoritmos e Estruturas de Dados III - 2017/01

Pedro Otávio Machado Ribeiro

14/06/2017

1 Introdução

Neste trabalho temos como contexto uma rua muito movimentada chamada Rua Pai no Bar, que, sugestivamente, possui muitos bares. Os donos dos bares desta rua moram no lado da rua oposto ao bar para que possam verificar facilmente qualquer movimentação suspeita. Para não falhar com a tradição, os donos dos bares querem pendurar bandeirolas para alegrar a vizinhança, já que a Copa na Rússia está chegando. Como os donos dos bares foram afetados pela crise, cada um fornecerá uma bandeirola que pode ser pendurada de sua casa até seu bar, porém uma bandeirola não pode se cruzar com a outra. Logo, dado uma lista de tamanho n de pares de números que representam o bar e a casa do dono, o objetivo deste trabalho é encontrar o maior número de bandeirolas que pode-se pendurar, respeitando as condições acima descritas. A númeração das ruas ocorre da forma usual: um lado com números pares, e o outro com números ímpares.

2 Metodologia

Como foi pedido, este problema foi solucionado por meio de 3 métodos diferentes: Força Bruta, algoritmo Guloso - que neste caso não se aplica, mas foi feita uma heurística para aproximar a resposta - e Programação Dinâmica.

Para representar um par de bar e seu dono, foi feita uma estrutura de dois números, em que o primeiro é o maior e o segundo, o menor. Logo, uma rua é representada por um vetor desta estrutura, e, antes de executar qualquer um dos métodos, este vetor é ordenado pelo primeiro elemento de cada estrutura. Desta forma, as bandeirolas ficam ordenadas pela posição em que ela termina.

Para detectar os cruzamentos, independentemente se o bar fica no lado ímpar ou par, sem perda de generalidade, basta comparar as coordenadas pares das bandeirolas e as ímpares da seguinte forma:

Seja $A = (p_a, i_a)$ e $B = (p_b, i_b)$ as bandeirolas representadas por um número par p e ímpar i que dizem a posição das extremidades em cada lado da rua. A e B não se cruzam se, e somente se, $(p_a > p_b \wedge i_a > i_b) \vee (p_a < p_b \wedge i_a < i_b)$.

Dado isso, segue abaixo a descrição da solução por meio de cada método.

2.1 Força Bruta

Para solucionar o problema com um algoritmo de força bruta, basta incluir ou não uma bandeirola. Dado que inicialmente o vetor de bandeirolas foi ordenado como descrito, se uma bandeirola b_i não se cruza com uma bandeirola b_{i-1}, então a bandeirola b_i não se cruzará nenhuma das bandeirolas b_k, onde $k \in [0, i-1[$. Dessa forma, uma bandeirola é adicionada à uma combinação se ela satisfaz essa condição, proporcionando assim, uma poda que impede que uma combinação com cruzamento seja formada.

2.2 Guloso

Dado que este problema não possui solução gulosa ótima, uma heurística que aproxima a solução foi desenvolvida.

A heurística feita para aproximar a solução é simples. Assuma que a primeira bandeirola faz parte da combinação solução de bandeirolas. Em seguida, percorre-se o vetor de bandeirolas, adicionando uma bandeirola se, e somente se ela não cruza com a última bandeirola adicionada na combinação, caracterizando assim uma escolha gulosa. O mesmo procedimento é realizado de trás pra frente, assumindo-se que o último faz parte da solução. Ao final, a combinação que tiver o maior tamanho é escolhida e definida como resposta.

3 Complexidade

3.1 Temporal

3.1.1 Força Bruta

O algoritmo de força bruta, na forma como foi feito, tem como pior caso a combinação em que todas as bandeirolas são inclusas, representado pelo último galho da árvore de recursão. Dessa forma, nenhuma poda acontecerá, e todas as 2^n combinações possíveis serão testadas, percorrendo assim todo o espaço de solução. Logo, a complexidade temporal do algoritmo de força bruta será $O(2^n)$.

3.1.2 Guloso

Dado que para obter a solução aproximada o vetor de bandeirolas é percorrido necessariamente duas vezes, assintóticamente o custo temporal da solução é $O(n)$.

3.2 Espacial

Todos os algoritmos desenvolvidos no máximo 3 vetores de tamanho máximo n. Logo, assintóticamente, o uso de memória é dado por $O(n)$.

4 Experimentos

Os testes foram realizados em meu computador pessoal, um notebook que não é mais um notebook, com as seguintes configurações:

- Sistema Operacional: Arch Linux
- Processador: Intel(R) Core(TM) i5-2430M CPU @ 4 $\times$ 2.40GHz
- Memória RAM: 6GB

Cada teste foi realizado 30 vezes e os valores apresentados são referentes a média destes.

Os casos de teste utilizados foram gerados por mim, utilizando um algoritmo que fornece uma entrada dado um tamanho n.

O tamanho dos casos de teste variam de 22 a 31 para o algoritmo de força bruta, de forma que o algoritmo sempre caia no pior caso, favorecendo assim uma melhor visualização dos dados. Segue abaixo o gráfico de tempo de execução:

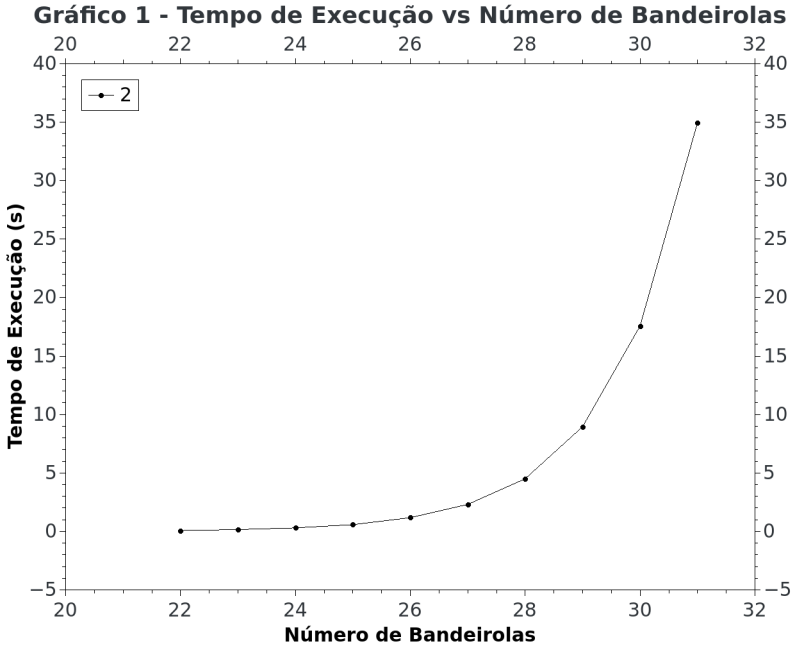

Figura 1: Tempo de Execução em função do Número de Bandeirolas para o força bruta

Para o algoritmo guloso, o tamanho dos casos de testes variam de 10000 a 14500, aumentando 500 a cada nova instância. Segue abaixo o gráfico de tempo de execução:

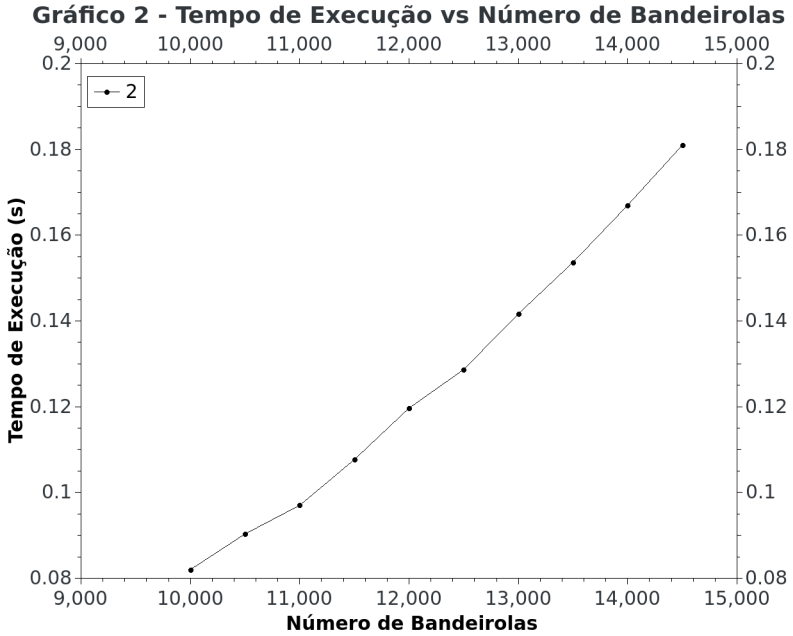

Figura 2: Tempo de Execução em função do Número de Bandeirolas para o guloso

5 Análise de Resultado

Nota-se que para o algoritmo de força bruta o gráfico demonstra um comportamento exponencial, bem como esperado dado que sua complexidade temporal é $O(2^n)$.

Para o algoritmo guloso, os testes mostram um comportamento linear conforme o tamanho da entrada aumenta, corroborando assim seu comportamento assintótico definido por $O(n)$.

6 Conclusão

Nota-se, por meio deste trabalho, que um mesmo problema pode ser solucionado utilizando paradigmas diferentes. Para este problema de pendurar bandeirolas, foram apresentados as soluções por meio de força bruta e guloso. O algoritmo força bruta fornece o resultado ótimo, porém este torna-se inviável de ser usado quando o número de bandeirolas é grande. Por outro lado, o algoritmo guloso, que neste caso é uma heurística que tenta se aproximar da resposta ótima mostra-se mais eficiente, no entanto ainda pode ser melhorada, dado que há outras características que podemos tirar vantagem, como a distância entre bandeirolas.