MC558: Projeto e Análise de Algoritmos II - 2s2022

Tarefa de Programação

Profº Flávio K. Miyazawa
PED: Elisa Dell'Arriva

Instituto de Computação - UNICAMP

Haroldo quer ser bruxo!

1 Descrição do Problema

Haroldo é um jovem garoto que deseja muito ser bruxo. Sendo assim, tão logo abriu uma escola de bruxos em sua cidade, ele já se matriculou e está ansioso para aprender. Como parte de seu treinamento, ele precisa aprender a fazer poções. Para realizar uma poção, são necessários um produto base e um conjunto de ingredientes específicos para a poção desejada. Haroldo está em sua casa e recebeu uma receita com a fórmula da poção que deve produzir. O produto base está em um laboratório da escola, que situa-se no outro lado da cidade. Para cumprir sua tarefa, ele precisa se deslocar de sua casa até o laboratório, coletando no caminho os ingredientes específicos que fazem parte da fórmula. Para tristeza de Haroldo, esse caminho não é contínuo, mas sim formado por vários pedaços isolados de terra firme e, a única forma de se mover de um pedaço a outro, é através de pontes que os ligam. Os ingredientes estão distribuídos por esse caminho de forma que cada pedaço de terra firme tem no máximo um ingrediente e cada ponte tem um fator de fragilidade. Para que Haroldo possa planejar uma rota, é lhe dado um mapa ilustrando a disposição dos pontos de terra firme, bem como o ingrediente, se houver, presente em cada ponto, e todas as pontes disponíveis, cada uma associada a uma probabilidade de não se romper. Sua missão é, dados um mapa e uma receita, planejar uma rota que sai da cada de Haroldo e chega no laboratório, coletando todos os ingredientes da receita e com maior chance de sucesso, isto é, maior chance de não ser interrompida devido à queda de pontes. É permitido passar por uma mesma ponte mais de uma vez, bem como visitar o mesmo pedaço de terra mais de uma vez.

Representação do mapa e da receita

O mapa e a receita são descritos em um arquivo de texto, organizado da seguinte maneira: A primeira linha contém dois inteiros n e m, os quais indicam o número de pedaços de terra e o número de pontes, respectivamente. Em seguida, vêm m linhas contendo três números u, v e p, indicando que existe uma ponte de u a v com chance p de não cair, sendo $u, v \in \{0, \ldots, n-1\}$ e $p \in [0, 1]$. A linha seguinte contém um inteiro k indicando o número de ingredientes que devem ser coletados. Cada ingrediente é representado por um inteiro de 1 a k. A próxima linha contém uma sequência $i_0 \ldots i_{n-1}$ de n números inteiros separados por um espaço, em que $0 \leq i_j \leq k$ representa o ingrediente presente no pedaço de terra j, com $0 \leq j < n$. Se $i_j = 0$, então não existe ingrediente no pedaço de terra j. As quantidades de pedaços de terra, de pontes e de ingredientes são limitadas por $2 \leq n \leq 100$, $1 \leq m \leq 10000$ e $1 \leq k \leq 6$, respectivamente. A casa de Haroldo fica no pedaço de terra 0 e o laboratório, no pedaço de terra $n-1$.

A saída deve ser um número $q \in [0, 1]$, com precisão de cinco casas decimais, indicando a probabilidade de sucesso da rota planejada. Caso não exista rota que saia da casa de Haroldo e chegue ao laboratório com todos os ingredientes, você deve imprimir 0 (zero). O valor de q será truncado exatamente uma vez e somente imediatamente antes de imprimi-lo.

Observações técnicas

O código deve ser feito em C++. É fornecido um arquivo template, no qual a leitura da entrada bem como a impressão da saída já estão implementadas. Além disso, o template contém uma função chamada `melhorRota`, a qual recebe os dados de entrada como parâmetros e deve retornar o valor encontrado pelo seu algoritmo. Note que esse valor é decimal. **Não arredondem ou trunquem esse valor antes de retorná-lo.** A questão da precisão será tratada no momento de impressão feita pelo código do template.

O intuito de fornecer esse template é justamente evitar problemas nas leitura e impressão dos valores. Você é livre para acrescentar funções auxiliares, definir tipos e incluir bibliotecas, se assim achar necessário. Você também pode optar por não utilizar o template, mas tenha em mente que, nesse caso, resultados errados por divergências de formatação e afins não serão reconsiderados.

2 Instruções para o Relatório

Você obrigatoriamente entregar um relatório sobre o desenvolvimento da tarefa. Para fins de clareza, o relatório deve ser estruturado em três seções:

- **1 Formulação e abordagem**
- **2 Descrição do código**
- **3 Análise de complexidade**

Na primeira seção, descreva como formulou o problema e a ideia adotada para resolução. Na segunda seção, forneça um resumo da organização do seu código, indicando também as estruturas de dados utilizadas e, finalmente, na terceira seção, apresente uma análise da complexidade de tempo do seu algoritmo.

3 Avaliação

A nota da tarefa será composta pela combinação da nota dada pelo SuSy, NS, com a nota do relatório, NR. A nota final da tarefa, NT, é calculada da seguinte forma:

$$NT = 0,8NS + 0,2NR.$$

A identificação de qualquer tentativa de plágio implicará em $NT = 0$ para todos os envolvidos, ou seja, a nota final da tarefa será zero.

4 Avisos importantes

- O código deve ser submetido no SuSy e o relatório deve ser submetido no classroom.
- A página da disciplina no SuSy é `https://susy.ic.unicamp.br:9999/mc558a`.
- Para submeter, utilizem somente os dígitos numéricos do RA e a senha da DAC.
- Espere, no mínimo, 60 segundos entre uma submissão e outra no SuSy.
- O número máximo de submissões no SuSy é 20 e somente a última será considerada para avaliação.
- Em caso de múltiplas submissões do relatório no classroom, somente a última será considerada para avaliação.
- **O prazo de entrega desta tarefa é 03/11/2022 às 23h59.** Submissões com atraso não serão consideradas para avaliação.