

Trabalho Prático 1 - Você está na vizinhança errada

1 Fique seguro

Você foi contratado como entregador em uma cidade conhecida por seus altos índices de criminalidade e você não quer correr riscos. Você, no entanto, recebe um mapa da cidade onde estão descritos os bairros e os trechos das ruas que os conectam. Além disso, no mapa estão as localizações das delegacias espalhadas pela cidade. Para cada trecho, existe também a probabilidade $P(u, v)$ de ser assaltado ao passar por um caminho entre o bairro u, v . Considere que $P(u, v) = P(v, u)$ e todas as ruas são de mão dupla.

Com base nessas informações, você decide que ao sair de um ponto de entrega para outro, você:

1. Nunca ficará a mais de k bairros de alguma delegacia durante o seu trajeto
2. Escolherá o caminho onde há a menor probabilidade de ser assaltado.

Seu trabalho é: dadas as informações que descrevem o mapa da cidade (com suas delegacias, ruas e probabilidades de ser assaltado), descubra o caminho que respeita as restrições descritas acima. Modele e mostre uma solução para o problema utilizando grafos.

2 Entrada

Você receberá arquivos de teste no seguinte formato: na primeira linha estará um número inteiro I que apresenta quantas instâncias o arquivo possui. Na próxima linha começa a primeira instância.

Na primeira linha do grafo contém o número de bairros Q , o número de ruas R que liga esses bairros, o número D que indica a quantidade de delegacias e o número K que é a qual distância você pode estar de alguma delegacia. A seguir, nas próximas R linhas estarão descritos os trechos entre bairros com dois inteiros - que representam os ids dos bairros, um número real positivo com precisão de duas casas decimais no intervalo $[0,1]$ que representa a probabilidade de ser assaltado ao se passar por aquele trecho. Nas próximas D linhas é mostrado quais bairros possuem uma delegacia. Na próxima linha existe um número N que representa o número de consultas do problema a serem simuladas. Nas próximas N linhas o id do ponto de saída S e chegada C . Assim que as N linhas terminarem o próximo grafo, caso exista mais de uma instância I , irá começar.

Veja o exemplo a seguir:

```

2
9 12 1 3
1 2 0.30
1 4 0.20
2 3 0.30
2 5 0.40
3 6 0.10
4 5 0.20
4 7 0.40
5 8 0.10
5 6 0.15
6 9 0.10
7 8 0.30
8 9 0.10
7
2
1 9
7 6
7 6 2 2
1 2 0.30
2 3 0.20
3 4 0.40
4 5 0.15
5 6 0.35
6 7 0.25
1
2
1
1 5

```

3 Saída

A saída para cada teste de entrada deve conter a probabilidade ser assaltado (com arredondamento de duas casas decimais) e o caminho a ser percorrido desde o quarteirão inicial até o quarteirão final que respeita as restrições já enunciadas. Para mais de uma consulta, da mesma instância deve se colocar cada saída em uma linha(sem deixar uma linha em branco). Quando tiver mais de uma instância as respostas devem conter um espaço em branco entre as instância.

A ordem de escolha da melhor rota deve respeitar a seguinte prioridade:

1. Menor probabilidade de ser assaltado
2. Ordem lexicográfica

Caso não seja possível resolver o problema seguindo as restrições enunciadas, a saída deverá conter apenas o número -1 .

Veja o exemplo de saída para o problema mostrado anteriormente:

-1

Fique atento à forma de calcular a probabilidade de ser assaltado. Note que o cálculo da probabilidade de ser assaltado no caminho $1 \rightarrow 4 \rightarrow 5$, da primeira instância, NÃO é dado por $P(1, 4) * P(4, 5)$ - isso representa a probabilidade de ser assaltado EM AMBOS os trechos. você quer saber a probabilidade de ser assaltado em qualquer um desses trechos.

Dado que $P_N(u, v) = 1 - P(u, v)$, sabemos que a probabilidade de não ser astaltado em qualquer trecho do caminho $(1, 4, 5)$ - que, por abuso de notação, chamaremos de $P_N(1, 4, 5)$ - é:

Portanto a probabilidade de ser assaltado durante o caminho $1 \rightarrow 4 \rightarrow 5$ é $1 - 0.64 = 0.36$.

- A data de entrega desse trabalho é **18/09/13**.
- A penalização por atraso obedece à seguinte fórmula $2^{d-1}/0.32\%$, onde d são os dias úteis de atraso.
- Submeta apenas um arquivo chamado <número matricula>_<nome>.zip. Não utilize espços no nome do arquivo. Ao invés disso utilize o caractere '_'.
'_'
- Não inclua arquivos compilados ou gerados por IDEs. **Apenas** os arquivos abaixo devem estar presentes no arquivo zip.
 - Makefile
 - Arquivos fonte (*.c e *.h)
 - Documentacao.pdf
- Não inclua **nenhuma** pasta. Coloque todos os arquivos na raiz do zip.
- Siga rigorosamente o formato do arquivo de saída descrito na especificação. Tome cuidado com whitespaces e formatação dos dados de saída
- **NÃO SERÁ NECESSÁRIO ENTREGAR DOCUMENTAÇÃO IMPRESSA!**

- Será adotada **média harmônica** entre as notas da **documentação e da execução**, o que implica que a nota final será 0 se uma das partes não for apresentada.

6 Documentação

A documentação não deve exceder 10 páginas e deve conter pelo menos os seguintes itens:

- Uma **introdução** do problema em questão.
- **Modelagem e solução** proposta para o problema. O algoritmo deve ser explicado de forma clara, possivelmente através de pseudo-código e esquemas ilustrativos.
- **Análise de complexidade** de tempo e espaço da solução implementada.
- **Experimentos** variando-se o tamanho da entrada e quaisquer outros parâmetros que afetem significativamente a execução.
- Especificação da(s) **máquina(s) utilizada(s)** nos experimentos realizados.
- Uma breve **conclusão** do trabalho implementado.

7 Código

O código deve ser obrigatoriamente escrito na **linguagem C**. Ele deve compilar e executar corretamente nas máquinas Linux dos laboratórios de graduação.

- O utilitário **make** deve ser utilizado para auxiliar a compilação, um arquivo *Makefile* deve portanto ser incluído no código submetido. O utilitário deverá gerar um executável com o nome **tp1** que deverá **obrigatoriamente** ser capaz de receber o nome de um arquivo de entrada - de onde serão lidas as instâncias do problema - e o nome de um arquivo de saída - onde serão gravadas as soluções. O comando para executá-lo deverá seguir o exemplo abaixo:
`./tp1 input.txt output.txt`
- As estruturas de dados devem ser **alocadas dinamicamente** e o código deve ser **modularizado** (divisão em múltiplos arquivos fonte e uso de arquivos cabeçalho .h)
- **Variáveis globais** devem ser evitadas.
- Parte da correção poderá ser feita de forma automatizada, portanto siga rigorosamente **os padrões de saída especificados**, caso contrário sua nota pode ser prejudicada.
- **Legibilidade e boas práticas** de programação serão avaliadas