TP1 - Biconectividade

17 pontos

Entrega: 07/04/2024

1 Objetivo do trabalho

Neste trabalho, vamos exercitar tópico relativos ao primeiro terço do curso: grafos e alguns algoritmos elementares nessas estruturas. Para tal, trabalharemos com uma generalização de um conceito visto em aula: *biconectividade*.

Serão fornecidos alguns casos de teste para que você possa testar seu programa, mas eles não são exaustivos! Podem haver situações que não são ilustradas por eles; cabe a você pensar em novos casos e garantir que seu programa esteja correto e que ele implemente um algoritmo de complexidade adequada.

1.1 Informações importantes

O código fonte do seu trabalho deve estar contido em um **único** arquivo na linguagem C++ e deve ser submetido via Moodle na tarefa `Entrega TP1` até o dia 07/04/2024. Você terá 30 tentativas para conseguir a nota total de execução; apenas a última submissão será levada em conta para fins de avaliação. Você não terá acesso a todos os casos de teste; determinar estratégias para testar seu programa e suas ideias faz parte do trabalho. Envios com atraso serão aceitos; leia a Seção 5 para a política de atrasos.

Plágio de qualquer natureza não será tolerado. Caso qualquer cola seja encontrada, seu trabalho será zerado e as demais providências cabíveis serão tomadas. Escreva seu próprio código, de maneira legível e com comentários apropriados; ele pode ser útil no futuro próximo.

2 Definição do problema

Era uma vez, em uma galáxia muito muito próxima da nossa, um pequeno planeta (os locais só o conhece como O Planeta) com uma população pacífica mas com constantes tempestades de raios e outros eventos eletromagnéticos. Devido às frequentes intemperes, as redes de computadores dO Planeta são todas cabeadas e novos cabos são caríssimos.

Infelizmente, é muito comum que um link de rede seja atingido por uma descarga eletromagnética e partes da rede se desconectem. Porém, contudo, toda via, entretanto, uma nova invenção está prestes a mudar tudo: o pára-raio. Mas para que essa tecnologia revolucionária seja devidamente instalada, uma única regra deve ser seguida:

1. Um pára-raio só pode ser instalado nos links de **borda** de um **cluster**, nunca no **interior** de um cluster.

NO Planeta, um cluster é definido como um conjunto **maximal** de links de rede que, mesmo que **exatamente um deles caia**, ainda é possível trocar mensagens entre todos os outros links da cidade. Um link está na **borda** de um cluster se ele pertence a mais de um cluster. O **interior** de um cluster é o conjunto de links de rede que só se comunicam com links do próprio cluster (ou seja, que não são links de borda). Note que um link pode pertencer a mais de um cluster diferente; veja, por exemplo, a Figura 1.

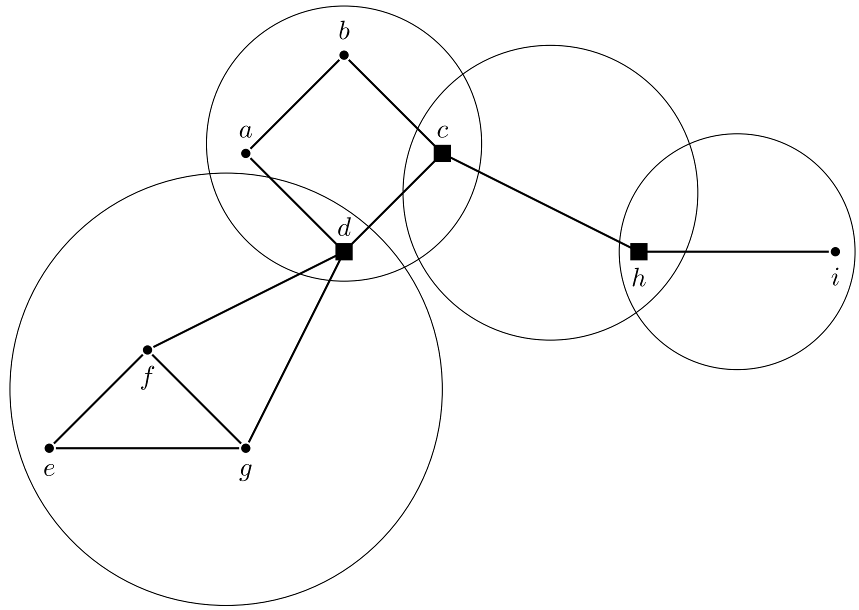

Figure 1: Uma possível configuração de rede dO Planeta. Os links denotados por quadrados são os possíveis candidatos a pára-raios; todos os outros links são interiores a algum cluster. Cada cluster é marcado por um círculo, sendo assim temos quatro clusters: $\{a, b, c, d\}, \{c, h\}, \{d, e, f, g\}, \{h, i\}$.

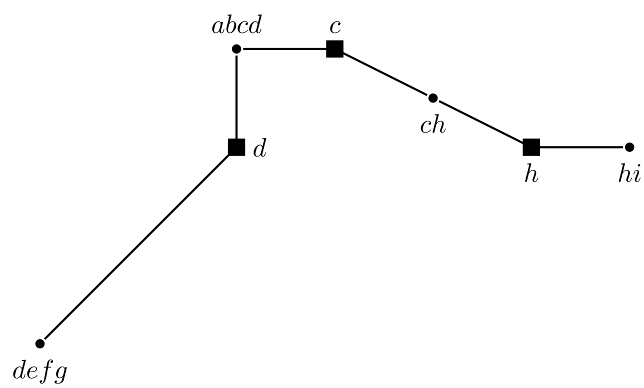

Figure 2: Floresta de clusters-bordas correspondente à rede da Figura 1

Além disso, a população dO Planeta está interessada em melhor mapear seus clusters. Para isso, eles querem construir a **Floresta de clusters-bordas** $\mathcal{T}$ da rede. Essa Floresta possui dois tipos de vértices: um vértice para cada cluster e um vértice para cada link de borda da rede original. Por fim, existe uma aresta entre um link de borda ℓ e um vértice de cluster C se e somente se o cluster correspondente a C contém o link de borda ℓ. Por exemplo, se o vértice de cluster C corresponde ao cluster $\{a, b, c, d\}$ da Figura 1 e estamos olhando para o vértice de borda d, então C e d são adjacentes em $\mathcal{T}$; por outro lado, se o vértice de borda for o vértice h, então C e h não são adjacentes.

Neste trabalho, dado um grafo G que representa uma rede dO Planeta você deve:

1. Listar todos os links de borda da rede;
2. Listar todos os clusters da rede;
3. Construir a Floresta de clusters-bordas da rede.

3 Dicas

Um segundo equivale a, aproximadamente, 10^8 operações em um computador moderno. Ou seja, um algoritmo $\mathcal{O}(N)$ para $N \leq 10^6$ cabe em um segundo, mas um algoritmo $\mathcal{O}(N^2)$ não cabe.

Operações de entrada e saída são extremamente caras: use as funções `scanf` e `printf` ao invés das funções `cin, cout`.

Links isolados não fazem parte de nenhum cluster!

4 Casos de teste

4.1 Formatado da Entrada

Cada caso de teste é composto por várias linhas, que representam uma rede G dO Planeta. A primeira linha contém dois inteiros, N e M, que representam, respectivamente, o número de links de rede e conexões entre eles. É garantido que $1 \leq N, M \leq 10^5$, e que $V(G) = \{1, 2, \ldots, N\}$. Seguem-se M linhas; a i-ésima dessas linhas contém dois inteiros x_i, y_i, que representam cada um, um link da rede G; é garantido que $x_i \neq y_i$, que nenhuma conexão é repetida, e que $x_i, y_i \in \{1, \ldots, N\}$.

4.2 Formato da Saída

Preste muita atenção a essa seção.

4.2.1 Listando os links de borda

A primeira linha da saída contém um inteiro F, que deve ser igual ao número de links de borda de G. As próximas F linhas devem corresponder, cada uma, a um link de borda de G. **Estes links devem estar ordenados pelo seu número**, ou seja, se $\{3, 1, 7\}$ são os links de borda de G, então eles devem ser impressos na ordem $\{1, 3, 7\}$.

4.2.2 Listando os clusters

A próxima linha da saída contém um único inteiro C, que deve ser igual ao número de clusters de G. Seguem-se, então, C linhas. A j-ésima dessas linhas começa com um inteiro z_j, **que deve ser igual a** $N + j$; esse é o identificador do cluster que será usado posteriormente para descrever a floresta de clusters-bordas. Na mesma linha, imprima também c_j, que representa o tamanho do j-ésimo cluster de G; ainda nessa linha, imprima c_j inteiros, que devem ser iguais aos links contidos no j-ésimo cluster. **Dentro de uma linha, os links devem estar ordenados pelo seu número, ou seja,** $\{3, 1, 7\}$ **será considerado incorreto, enquanto** $\{1, 3, 7\}$ **será considerado correto. Clusters devem ser ordenados lexicograficamente. Ou seja, se temos os clusters** $a = \{1, 3, 7\}$ **e** $b = \{1, 2, 100\}$**, o cluster** b **deve ser listado** ***antes*** **do cluster** a**, e, portanto, possuir um identificador menor.**

4.2.3 Construindo a Floresta

A próxima linha da saída deve conter dois inteiros Z e L, que devem corresponder ao número de vértices da Floresta de clusters-bordas e ao número de arestas dela, respectivamente. Seguem-se, então, L linhas. A k-ésima dessas linhas deve conter exatamente dois inteiros x_k, y_k, que representam dois vértices da Floresta. **Você deve garantir que: (i)** $x_k < y_k$**, e (ii) as arestas da Floresta aparecem em ordem lexicográfica, ou seja,** 113 **tem de vir depois de** 14**.** Para vértices que correspondem a clusters **você deve utilizar os identificadores gerados na etapa** ***Listando os clusters*****.**

4.3 Limites de execução

Para qualquer caso de teste, seu código deve imprimir a resposta em, no máximo, 3 segundos. Seu programa deve usar menos de 100MB de memória. Estruturas de dados devem ser alocadas sob demanda; ou seja, não faça vetores estáticos gigantescos para grafos com poucos vértices. Todas as avaliações serão feitas automaticamente via VPL. Programas que não aderirem a essas restrições para um teste terão a nota do mesmo zerada.

Lembre-se: você pode submeter uma solução para a tarefa no máximo 30 vezes e apenas a última submissão será levada em conta para fins de avaliação.

4.4 Exemplos

4.4.1 Exemplo da Figura 1

No exemplo abaixo, o vértice de número i corresponde à i-ésima letra do alfabeto;, i.e. $4 = d$.

Entrada	Saída
9 11	3
1 2	3
2 3	4
3 4	8
3 8	4
4 1	10 4 1 2 3 4
4 6	11 2 3 8
4 7	12 4 4 5 6 7
5 6	13 2 8 9
5 7	7 6
6 7	3 10
8 9	3 11
	4 10
	4 12
	8 11
	8 13

4.4.2 Exemplo 2

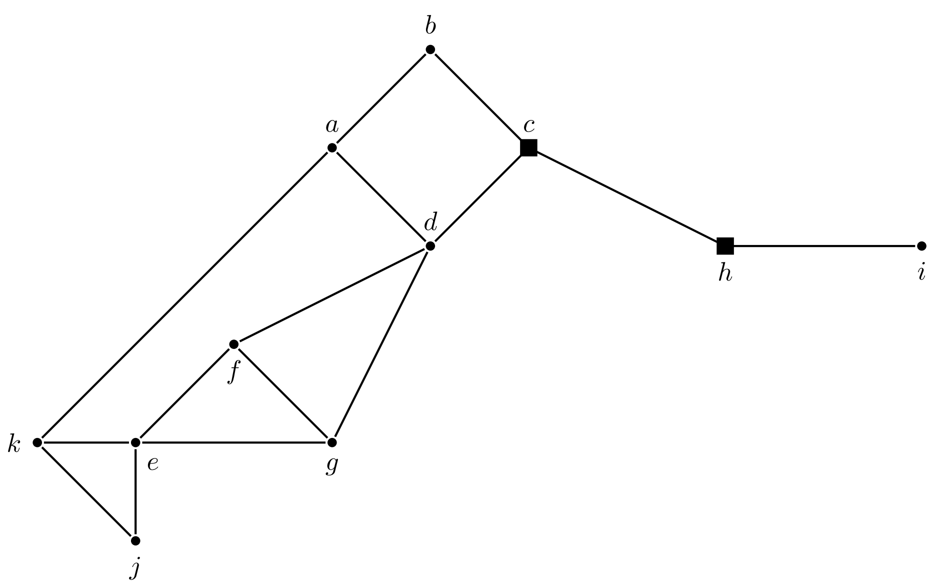

Figure 3: Uma possível configuração de rede dO Planeta. Os links denotados por quadrados são os possíveis candidatos a pára-raios; todos os outros links são interiores a algum cluster. Temos três clusters: $\{a, b, c, d, e, f, g, j, k\}, \{c, h\}, \{h, i\}$.

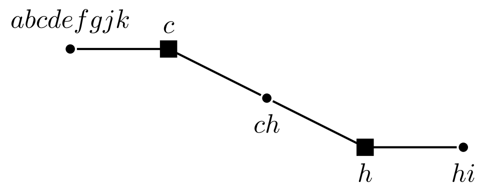

Figure 4: Floresta de clusters-bordas correspondente à rede da Figura 3

No exemplo abaixo, o vértice de número i corresponde à i-ésima letra do alfabeto;, i.e. $4 = d$.

Entrada	Saída
11 15	2
1 2	3
2 3	8
3 4	3
3 8	12 9 1 2 3 4 5 6 7 10 11
4 1	13 2 3 8
4 6	14 2 8 9
4 7	5 4
5 6	3 12
5 7	3 13
6 7	8 13
8 9	8 14
5 10	
5 11	
10 11	
1 11	

5 Atrasos

O trabalho pode ser entregue com atraso, mas será penalizado de acordo com a seguinte fórmula, onde d é o número de dias atrasados:

$$\Delta(d) = \frac{2^{d-1}}{0.32}\% \tag{1}$$

Por exemplo, com um atraso de quatro dias e uma nota de execução de 70% do total, sua nota final será penalizada em 25%, ficando assim igual a $70 \cdot (1 - \Delta(d)) = 52.5\%$. Note que a penalização é exponencial, e um atraso de 6 dias equivale a uma penalidade de 100%.

6 Errata

- Correção no formato de saída.
- Adição do exemplo 2.
- Dica dos links isolados.