

MAC 6711 - Tópicos de Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2020

Lista 5

1. **(KT, Cap 4, Ex 8)** Considere um grafo G conexo, com pesos distintos nas arestas. Prove que G tem uma única árvore geradora mínima.
2. **(KT, Cap 4, Ex 9)** O problema da árvore geradora mínima tem como objetivo encontrar uma árvore geradora cujo custo total é o menor possível. Vamos explorar agora um outro objetivo: encontrar uma árvore geradora para a qual a *aresta mais cara* é o mais barata possível. Especificamente, seja $G = (V, E)$ um grafo conexo com n vértices, m arestas e custos positivos nas arestas, todos distintos. Seja $T = (V, E')$ uma árvore geradora de G ; defina a aresta *gargalo* como sendo a aresta de T com o maior custo. Uma árvore geradora é uma *árvore geradora de gargalo mínimo* se não existe nenhuma outra árvore geradora de G com aresta gargalo mais barata.
 - (a) Toda árvore geradora de gargalo mínimo é uma árvore geradora mínima? Prove ou dê um contra-exemplo.
 - (b) Toda árvore geradora mínima é uma árvore geradora de gargalo mínimo? Prove ou dê um contra-exemplo.
3. **(KT, Cap 4, Ex 10)** Seja $G = (V, E)$ um grafo com um custo $c_e \geq 0$ para cada aresta $e \in E$. Considere uma árvore geradora T de custo mínimo em G . Suponha que uma aresta nova é adicionada a G , conectando dois vértices v e w de V , com custo c .
 - (a) Descreva um algoritmo eficiente para testar se T continua uma árvore geradora mínima com a nova aresta adicionada a G . Seu algoritmo deve consumir tempo $O(|E|)$. Você consegue descrever um algoritmo que consuma tempo $O(|V|)$? Deixe claro qualquer hipótese que você assuma sobre a ED que estaria sendo usada para armazenar o grafo G e a árvore T .
 - (b) Suponha que, com a adição da aresta, T não é mais uma árvore geradora mínima. Descreva um algoritmo linear, ou seja, $O(m)$, para atualizar a árvore T , transformando-a numa árvore geradora mínima do grafo corrente.
4. **(KT, Cap 4, Ex 11)** Considere um grafo $G = (V, E)$ conexo, com um custo c_e para cada aresta $e \in E$. No exercício 1, vimos que, se todos os custos forem distintos, então G tem uma única árvore geradora mínima. No entanto, G pode ter várias árvores geradoras mínimas quando os custos não são todos distintos. Aqui formulamos uma pergunta: podemos ajustar o algoritmo de Kruskal para que ele encontre todas as árvores geradoras mínimas de G ?

Lembre-se que o algoritmo de Kruskal ordena as arestas pelo seu custo, e então gulosamente processa uma aresta por vez nesta ordem, adicionando à floresta corrente uma aresta e sempre que ela não forma um circuito com as já escolhidas. Quando existem arestas com o mesmo custo, o algoritmo não especifica em que ordem tais arestas devem ser examinadas. Qualquer ordem entre elas é válida. Considere uma árvore geradora mínima arbitrária de G . É verdade que sempre existe uma ordem válida das arestas de G para a qual o algoritmo de Kruskal produz como saída essa árvore? Prove ou dê um contra-exemplo.

5. **(KT, Cap 4, Ex 19)** Um grupo de projetistas de rede da companhia de comunicação CluNet encara o seguinte problema. Eles têm um grafo $G = (V, E)$ conexo, no qual os nós representam locais que querem se comunicar. Cada aresta e representa um cabo de comunicação, com uma dada largura de comunicação b_e .

Para cada par de nós $u, v \in V$, eles querem selecionar um uv -caminho P pelo qual u e v vão se comunicar. O *gargalo* $b(P)$ deste caminho P é a largura mínima dentre as arestas de P , ou seja, $b(P) = \min\{b_e : e \in P\}$. O *melhor gargalo atingível* para o par $u, v \in V$ é simplesmente o máximo, sobre todos os uv -caminhos P de G , do valor $b(P)$.

É complicado manter um caminho para cada par de nós de G , assim um dos projetistas fez uma sugestão ousada: talvez pudéssemos encontrar uma árvore geradora T de G tal que, para *todo* par de nós u, v , o único uv -caminho na árvore T atinge o melhor gargalo para u, v em G . (Em outras palavras, mesmo que você pudesse escolher qualquer uv -caminho no grafo todo, você não conseguiria um uv -caminho melhor que o uv -caminho de T .)

Esta ideia foi discutida intensamente nos escritórios da CluNet por alguns dias, porém há naturalmente um certo ceticismo: cada par de nós pode querer um caminho completamente diferente para maximizar o seu gargalo; por que existiria uma árvore que simultaneamente fizesse todos os pares felizes? Mas, depois de várias tentativas fracassadas de achar um furo nessa ideia, as pessoas começaram a achar que ela poderia estar correta.

Mostre que uma tal árvore existe, e descreva um algoritmo eficiente para encontrá-la, ou seja, descreva um algoritmo que, dado G e b , constrói uma árvore geradora T na qual, para cada $u, v \in V$, o gargalo do uv -caminho em T é igual ao melhor gargalo atingível para o par u, v em G .

6. **(KT, Cap 4, Ex 20)** Todo outono, em alguma parte fria e montanhosa do mundo, os policiais rodoviários se reúnem e decidem quais estradas manterão limpas no próximo inverno. A região consiste de n cidades, e o sistema de estradas pode ser visto como um grafo conexo $G = (V, E)$ onde V é o conjunto de cidades e cada aresta representa uma das estradas ligando duas cidades. No inverno, as pessoas, por estarem na parte bem alta da montanha, param de se preocupar com o *comprimento* das estradas e começam a se preocupar com a sua altitude — isso é o que determina quão difícil será a viagem.

Então cada estrada — cada aresta e do grafo — está associada a um número a_e que representa a altitude máxima da estrada. Assuma que não há duas arestas com a mesma altitude máxima a_e . A *altura* de um caminho P no grafo é então o máximo dos a_e sobre todas as arestas e em P . Finalmente, um caminho entre duas cidades i e j é declarado *inverno-ótimo* se ele atinge a menor altitude possível entre todos os ij -caminhos no grafo.

Os policiais rodoviários estão escolhendo um conjunto $E' \subseteq E$ das estradas para manter limpas durante o inverno; o restante das estradas serão deixadas sem manutenção e fechadas para os viajantes durante o inverno. Todos eles concordam que, qualquer que seja o subconjunto das estradas E' que eles decidam manter limpas, ele deve ser tal que o subgrafo (V, E') seja conexo; e mais fortemente, para todo par de cidades i e j , a altitude do caminho inverno-ótimo no grafo (V, E') não deve ser maior que a de um caminho inverno-ótimo no grafo $G = (V, E)$ todo. Diremos que (V, E') é um *subgrafo conexo de altitude mínima* se ele tem essa propriedade.

Embora eles queiram manter essa propriedade chave, eles por outro lado querem manter também o número menor possível de estradas limpas. Então um belo dia eles se deparam com a seguinte conjectura:

A árvore geradora mínima de G , com respeito aos custos a_e , é um subgrafo conexo de altitude mínima.

(No primeiro exercício, mostramos que, se os custos nas arestas são distintos, existe uma árvore geradora mínima única. Assim, graças à hipótese de que os a_e 's são distintos, podemos falar *da* árvore geradora mínima.)

Inicialmente essa conjectura é de alguma maneira contra-intuitiva, já que a árvore geradora mínima está tentando minimizar a *soma* dos valores a_e , enquanto que o objetivo de minimizar a altitude parece estar pedindo algo razoavelmente diferente. Mas, por falta de um argumento contrário, os policiais começaram a considerar uma segunda conjectura ainda mais ousada:

Um subgrafo (V, E') é um subgrafo conexo de altitude mínima se e somente se ele contém as arestas da árvore geradora mínima.

Note que esta segunda conjectura implica na primeira, já que a árvore geradora mínima contém suas arestas.

Então aqui está a pergunta.

- (a) A primeira conjectura é verdadeira para todo grafo G e toda escolha de altitudes a_e distintas? Prove ou dê um contra-exemplo.
 - (b) A segunda conjectura é verdadeira para todo grafo G e toda escolha de altitudes a_e distintas? Prove ou dê um contra-exemplo.
7. **(KT, Cap 4, Ex 22)** Considere o problema da árvore geradora mínima num grafo $G = (V, E)$ com um custo $c_e \geq 0$ para cada aresta e , onde os custos podem não ser todos distintos. Se os custos não são todos distintos, podem existir em geral várias árvores geradoras mínimas diferentes. Considere uma árvore geradora $T \subseteq E$ com a garantia que, para todo $e \in T$, e pertence a alguma árvore geradora mínima de G . Podemos concluir que T é uma árvore geradora de custo mínimo de G ? Prove ou dê um contra-exemplo.
8. Considere a implementação do union-find por árvores enraizadas. Escreva uma versão não recursiva do FINDSET com compressão de caminhos.
9. Considere a implementação do union-find por árvores enraizadas. Prove que o custo de pior caso de uma chamada do FINDSET(x) é $O(\lg n)$, onde n é o número de elementos na árvore onde está o x .