MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2015

Lista 8

1. **(CRLS Ex. 23.1-1)** Seja e uma aresta de custo mínimo em um grafo G com custos nas arestas. É verdade que e pertence a alguma MST de G? É verdade que e pertence a toda MST de G?

2. Suponha que os custos das arestas de um grafo conexo são distintos dois a dois (ou seja, não há duas arestas com o mesmo custo). Mostre que o grafo tem uma única MST.

3. Suponha que os custos das arestas de um grafo conexo são distintos dois a dois. Seja C um ciclo não trivial. É verdade que a aresta de custo mínimo em C pertence à (única) MST do grafo?

4. **(CRLS Ex. 23.1-2)** Prove ou desprove a seguinte afirmação: Dado um grafo G com pesos nas arestas, um conjunto de arestas A de G, e um corte que respeita A, toda aresta que cruza o corte e que é segura para A tem peso mínimo dentre todas as arestas desse corte.

5. **(CRLS Ex. 23.1-3)** Prove ou desprove a seguinte afirmação: Se uma aresta está contida em alguma MST, então tem peso mínimo dentre todas as arestas de algum corte no grafo.

6. **(CRLS Ex. 23.1-7)** Prove que se todos os pesos nas arestas são positivos, então qualquer subconjunto de arestas que conectam todos os vértices e tem peso total mínimo forma uma árvore. A propriedade vale se alguns pesos são negativos?

7. Seja T uma MST de um grafo com pesos positivos e distintos nas arestas. Suponha que substituímos cada peso pelo seu quadrado. Verdadeiro ou falso: T ainda é uma MST para o novo grafo.

8. Dado um grafo conexo G, dizemos que duas MSTs T e T' são vizinhas se T contém exatamente uma aresta que não está em T', e T' contém exatamente uma aresta que não está em T. Vamos construir um novo grafo (muito grande) $\mathcal{H}$ como segue. Os vértices de $\mathcal{H}$ são as MSTs de G, e existe uma aresta entre dois vértices em $\mathcal{H}$ se os correspondentes MSTs são vizinhas. É verdade que $\mathcal{H}$ é sempre conexo? Prove ou dê um contra-exemplo.

9. Seja G um grafo conexo com custos nas arestas. Uma aresta e de G é crítica se o aumento do custo de e faz com que o custo de uma MST de G também aumente. Escreva uma função que determine todas as arestas críticas de G em tempo $\mathrm{O}(m \log n)$

10. Mostre que depois de cada execução da linha 12 do algoritmo de Prim tem-se $\text{key}[u] < \infty$

11. Suponha que temos um grafo G com pesos nas arestas. Verdadeiro ou falso: Para qualquer MST T de G, existe uma execução válida do algoritmo de Kruskal que produz T como saída? Dê uma prova ou um contra-exemplo.

12. Seja G um grafo conexo com custos nas arestas e seja B um conjunto de arestas de G. Suponha que o grafo induzido por B não tem circuitos. Queremos encontrar uma subárvore geradora de custo mínimo dentre as que contêm B. Descreva um algoritmo eficiente para resolver o problema.

13. **(CRLS Ex. 23.2-4,5)** Suponha que todos os pesos num grafo com n vértices são inteiros no intervalo de 1 até n. Descreva como otimizar os algoritmos de Kruskal e Prim nesta situação. O que acontece se os pesos são inteiros no intervalo de 1 até W?

14. Dado um grafo com n vértices, pesos distintos nas arestas, e no máximo $n + 8$ arestas, dê um algoritmo com complexidade $\mathrm{O}(n)$ para achar uma MST.

15. O diâmetro de um grafo é o máximo das distâncias entre dois vértices. Escreva código que usa o algoritmo de Dijkstra para calcular o diâmetro de um grafo.

16. Considere um digrafo (grafo orientado) com custos positivos associados aos vértices. O custo de um caminho num tal digrafo é a soma dos custos dos vértices do caminho. Queremos encontrar um caminho de custo mínimo dentre os que começam num vértice s e terminam num vértice t. Adapte o algoritmo de Dijkstra para resolver esse problema.

17. Sejam s e t dois vértices de um digrafo com custos positivos nos arcos. Para cada vértice v do digrafo, seja $x[v]$ o custo de algum caminho de s a v. Escreva um algoritmo eficiente que verifique se $x[t]$ é a distância de s a t em G. Explique porque seu algoritmo está correto.

18. Mostre que o algoritmo de Dijkstra pode produzir resultados errados se o digrafo tiver arcos de custo estritamente negativo.

19. Escreva um algoritmo que recebe conjuntos S e T de vértices de um grafo e calcula a distância de S a T, ou seja, o custo de um caminho de custo mínimo que começa em algum vértice em S e termina em algum vértice em T. O algoritmo deve consumir o mesmo tempo de execução que o algoritmo de Dijkstra. Justifique que seu algoritmo está correto. *Dica:* Basta introduzir uma pequena modificação no algoritmo de Dijkstra.

20. Escreva um algoritmo que encontre um arco cuja remoção causa o maior aumento na distância de um vértice s a um vértice t.

21. Suponha que trocamos a linha 4 do algoritmo do Dijkstra como segue

$$4.\text{while}|Q| > 1$$

Isso faz com que a execução do laço execute $|V|-1$ vezes no lugar de $|V|$ vezes. Será que o algoritmo continua correto?

22. Dado um digrafo $G = (V, E)$ em que cada aresta $(u, v) \in E$ tem associado um valor $r(u, v)$, que é um número real no intervalo $[0, 1]$ que representa a confiança de um canal de comunicação do vértice u até o vértice v. Interpretamos $r(u, v)$ como a probabilidade de que o canal de u a v não falhe, e supomos que tais probabilidades são independentes. Dê um algoritmo eficiente (mesmo tempo de execução que o de Dijkstra) que acha um caminho mais confiável entre dois vértices dados.

23. Seja $G = (V, E)$ um digrafo com pesos $w : E \to \{0, 1, \dots, W\}$ para algum W. Modifique o algoritmo de Dijkstra para que compute os caminhos mínimos a partir de um vértice s em tempo $\mathrm{O}(W|V| + |E|)$.

24. Seja $G = (V, E)$ um digrafo com pesos $w : E \to \{0, 1, \dots, W\}$ para algum W. Modifique o algoritmo de Dijkstra para que compute os caminhos mínimos a partir de um vértice s em tempo $\mathrm{O}((|V| + |E|) \lg W)$. (*Dica:* Quantos caminhos máximos distintos podem exister em $V - S$ em cada iteração do algoritmo?)