MAC0338 - ANÁLISE DE ALGORITMOS

LISTA 7

FIZ: 7 / 14

CHECKLIST

entregue   fiz   incompleto   não entendi   CRLS

- árvore geradora mínima

1   2   3   4   5   6   7

8   9

- algoritmo de Prim

10

- algoritmo de Kruskal

11

- Kruskal e Prim

12   13   14

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?

PROVA POR CONTRADIÇÃO

"É verdade que e pertence a alguma MST de G?"

- Seja um grafo G, com V vértices e E arestas
- suponha uma MST T sem a aresta e
- a MST tem V-1 arestas e não tem ciclos.
- agora, adicione a aresta e na MST
- será possível notar que um ciclo se formará devido à adição de e
- o ciclo inclui a aresta e e outras arestas que fazem parte de T
- neste ciclo, selecione uma aresta cujo peso seja o maior, certamente não será a aresta e, pois e tem o menor peso.
- portanto, será selecionada uma aresta da MST T anterior
- seja e1 essa aresta.
- remova e1 e adicione e na MST T.
- seja T1 essa nova MST.

- Quando removemos uma aresta no ciclo, a conectividade da árvore é mantida
- além disso, como ainda existem V-1 arestas, não há ciclos em T1.
- portanto, T1 é uma MST.

- peso (T1) = peso (T) - (e1 - e)
- (e1 - e) é positivo, porque e é mínimo.
- portanto, peso (T1) < peso (T)
- e T não pode ser uma árvore geradora mínima que possui e

- Assim, existe uma MST com e.
- ainda mais, e pertence a toda MST de G

(caso todas as arestas tenham peso único)

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.

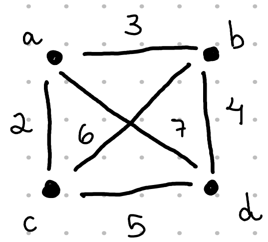
If T_1 and T_2 are *distinct* minimum spanning trees, then consider the edge of minimum weight among all the edges that are contained in *exactly one* of T_1 or T_2. Without loss of generality, this edge appears only in T_1, and we can call it e_1.

Then $T_2 \cup \{e_1\}$ must contain a cycle, and one of the edges of this cycle, call it e_2, is not in T_1.

Since e_2 is a edge different from e_1 and is contained in *exactly one* of T_1 or T_2, it must be that $w(e_1) < w(e_2)$. Note that $T = T_2 \cup \{e_1\} \setminus \{e_2\}$ is a spanning tree. The total weight of T is smaller than the total weight of T_2, but this is a contradiction, since we have supposed that T_2 is a minimum spanning tree.

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?

Suponha o seguinte ciclo C



a aresta de peso mínimo é (a,c) com custo 2

PARECIDO

COM A

$\boxed{1}$

Sabrina Araújo da Silva nUSP 12566182

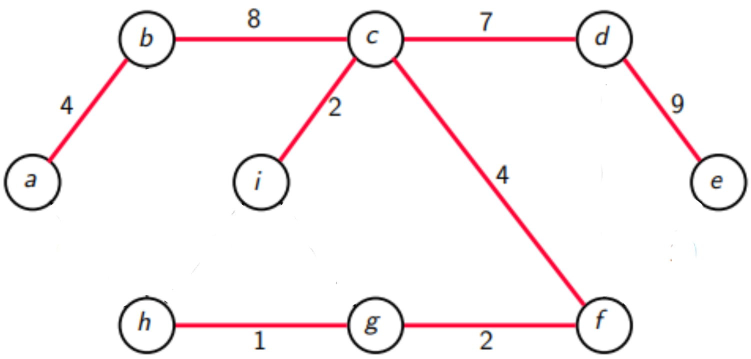
MAC0338 - ANÁLISE DE ALGORITMOS

LISTA 7

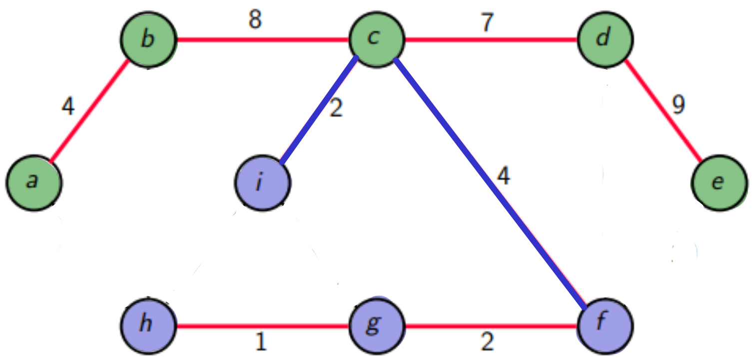
exercícios 4, 11 e 13

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.

Seja G o seguinte grafo.



Agora, seja A o conjunto de arestas {(a,b), (b,c), (c,d), (d,e)}, destacado em verde. E um corte com os vértices {f, g, h, i}, destacado em azul.

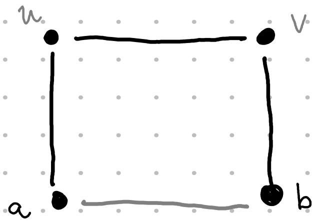


As arestas que cruzam o corte são: (c,i) e (c,f). Como G é a sua própria MST, as arestas mencionadas são seguras, pois estão na MST de G. Porém, essas arestas não tem peso mínimo dentre todas as arestas do corte. Podemos observar que o custo da aresta (h,g) é menor que o custo das arestas (c,i) e (c,f). E até mesmo dentre as arestas que cruzam o corte, (c,f) não tem peso mínimo.

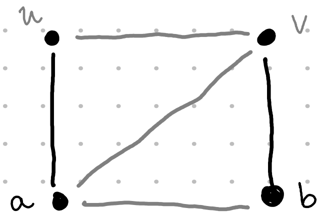
Portanto, a afirmação não é verdadeira.

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.

- Seja uma aresta (u,v) em uma MST.



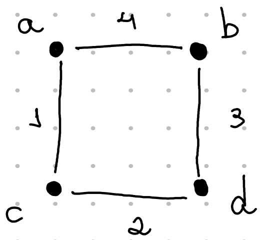
- Agora, considere retirá-la da MST



- Sejam dois conjuntos $A=(u,a)$ e $B=(v,b)$ que se formaram ao retirar a aresta (u,v).
- Dentre as arestas que cruzam o corte, (u,v) é a que possui custo mínimo. Caso contrário, (a,v) ou (a,b) iriam fazer parte da MST, contradizendo o fato que a a árvore que contém (u,v) é uma MST.

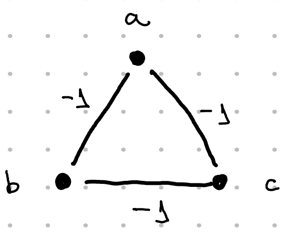
6. **(CRLS Ex. 23.1-7)** Prove que se todos os pesos nas arestas são positivos, então qualquer subconjunto de arestas que conectam todos os vértices e tem peso total mínimo forma uma árvore. A propriedade vale se alguns pesos são negativos?

Seja G o seguinte grafo



- é visível que todos os vértices formam caminho até outro vértice. Agora suponha que (a,b) é a aresta com maior custo, ao retirá-la do ciclo todos os vértices ainda estarão conectados, porém o peso total das arestas será menor e o grafo não formará um ciclo. Portanto, o subconjunto restante de vértices forma uma árvore de peso mínimo.

- Agora, suponha o seguinte grafo G' no qual suas arestas têm peso -1



- todos os vértices estão conectados e tem peso total -3.
- Ao retirar arbitrariamente o vértice (a,b), teremos uma árvore com peso total -2.
- Assim, contradizendo que qualquer conjunto de arestas com peso mínimo forma uma árvore. Portanto, a propriedade não vale para pesos 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.

T ainda é uma MST para o novo grafo, porque para encontrar uma MST apenas comparamos os seus pesos. Como o peso de cada uma das arestas foi aumentado proporcionalmente em relação à todas, temos que todos os resultados das comparações feitas para formar T se mantém.

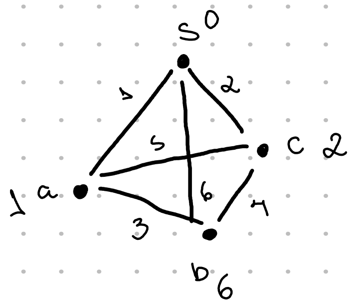
8. Dado um grafo conexo G, dizemos que duas MSTs T e T' são vizinhas se T contém exatamente uma aresta que não está em T', e T' contém exatamente uma aresta que não está em T. Vamos construir um novo grafo (muito grande) $\mathcal{H}$ como segue. Os vértices de $\mathcal{H}$ são as MSTs de G, e existe 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 $O(m \log n)$

Usar Kruskal para descobrir as arestas de peso igual e analisar se vão ser usadas ou não, se não forem então não são críticas

10. Mostre que, se G é conexo, depois da remoção de u da fila Q na linha 6 do algoritmo de Prim, tem-se $\text{key}[u] < \infty$.

Se G é conexo, há pelo menos um caminho entre qualquer par de vértices. Então, para cada vértice há pelo menos um vértice no qual ele é adjacente. Portanto, antes de extrair u, ele é encontrado por um vértice adjacente e, a partir disso, a key desse vértice sempre será menor que infinito.



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.

O algoritmo de Kruskal percorre uma sequência ordenada de arestas, as selecionando uma a uma para formar a árvore geradora mínima. Se cada aresta do grafo G tiver peso diferente das outras, o grafo terá uma MST única. Porém, se arestas diferentes tiverem o mesmo peso, o grafo terá mais de uma MST.

Neste último caso, o algoritmo de Kruskal escolhe uma dessas arestas com peso igual de acordo com algum critério guloso. Para obter determinada MST T de G, é possível adicionar algum critério para que a ordem no qual as arestas serão exploradas conduza para uma MST específica. Esse critério, por exemplo, pode ser algum peso arbitrário adicional às arestas com peso igual. Assim, o algoritmo de Kruskal poderá produzir qualquer árvore geradora mínima, apenas dependendo da ordem que as arestas serão buscadas. Portanto, a afirmação é verdadeira.

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.

```
Prim (G, c, s, B
    para v ∈ V(G) faça Key[v] ← ∞
    Key[s] ← 0     π[s] ← nil
    Q ← V(G)
    enquanto Q ≠ ∅ faça
        u ← extract_min (Q)
        para v ∈ adj(u) faça
            se v ∈ Q e (uv) ∈ B e c(uv) < Key[v]
                então π[v] ← u e Key[v] ← c(uv)
    devolva π
```

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?

Para o algoritmo de Kruskal:

- se todos os pesos no grafo com n vértices são inteiros no intervalo de 1 até n, podemos ordernar as arestas usando o counting sort, que se baseia na ordenação de uma lista de inteiros quando se sabe que os inteiros estão em um certo intervalo. Essa ordenação terá complexidade $O(n+m)$, no qual n é o tamanho do intervalo e m o número de arestas.
- se os pesos são inteiros no intervalo de 1 até W, podemos usar a mesma estratégia anterior. A ordenação do counting sort terá complexidade $O(W+m)$, no qual W é o tamanho do intervalo e m o número de arestas.

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.

usa DFS

para cada aresta incidente em s

Kruskal (G, c)

```
A ← ∅
sejam e1 ... em as arestas ordenadas por c
para cada u ∈ V(G) faça MakeSet(u)
para i ← 1 até m faça
    sejam u e v as pontas de ei
    se FindSet(u) ≠ FindSet(v)
        então A ← A ∪ {ei}
            Union(u, v)
devolva A
```