Universidade Federal de Pernambuco
Centro de Informática
Bach. em Engenharia da Computação

IF672ec Algoritmos e Estr. de Dados 2016.1
Prof. Paulo Fonseca

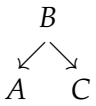
Segunda prova — 30 de Junho de 2016

QUESTÃO 1 (2,0pt)

Num percurso em profundidade de um grafo (DFS), se seguimos uma aresta para visitar um nó w a partir de um nó v, então a aresta (v, w) é dita uma *aresta de árvore*. A união de todas as arestas de árvore de uma DFS forma uma árvore geradora do grafo (árvore DFS). Se, por outro lado, ao visitar um nó v, tivermos uma aresta (v, u) para um nó u já marcado, então essa aresta é dita uma *aresta de retorno*.

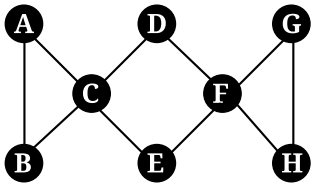
Um grafo é dito *biconexo* se ele é conexo e continua conexo se removermos um vértice qualquer. Um vértice v que desconecta o grafo se for removido é chamado *ponto de articulação*. Um percurso DFS pode ser utilizado para detectar pontos de articulação num grafo, conforme descrito a seguir.

Considere uma árvore DFS de um percurso iniciado num vértice s. Se o nó s, que é a raiz da árvore DFS, possui mais de um filho, então s é um ponto de articulação. Por exemplo, considere o grafo $A - B - C$, no qual B é claramente um ponto de articulação. Se iniciarmos a DFS em B, então de fato teremos a árvore DFS



, ou seja B tem dois filhos. Se um nó v não é a raiz da árvore DFS, então ele será um ponto de articulação se possui um filho w tal que nenhum nó da subárvore enraizada em w tem uma aresta de retorno para um ancestral de v na árvore DFS. No mesmo exemplo anterior, se começarmos a DFS em A, teremos a árvore DFS $A \rightarrow B \rightarrow C$, ou seja B tem um filho C tal que nenhum nó da árvore enraizada nele tem aresta de retorno para A, que é o ancestral de B.

Ilustre a detecção dos pontos de articulação do grafo a seguir representando a árvore DFS a partir do vértice C. Represente as arestas de árvore por linhas cheias ($\rightarrow$), e as arestas de retorno por linhas pontilhadas ($\dashrightarrow$). Indique claramente os pontos de articulação na árvore.



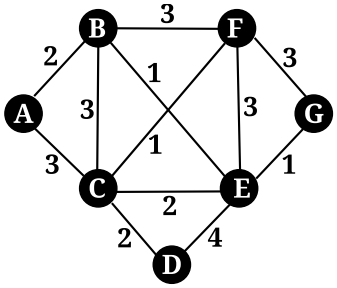
QUESTÃO 2 (3,0pt)

a) Complete o diagrama

Iter. #	Peso, Precursor A	B	C	D	E	F	G
0	$0, -$	$\infty, ?$	$\infty, ?$	$\infty, ?$	$\infty, ?$	$\infty, ?$	$\infty, ?$
1	$(0, -)$	$2, A$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

correspondente à execução do *Algoritmo de Dijkstra* sobre o grafo a seguir.

b) As arestas que ligam cada vértice ao seu precursor formam uma árvore geradora. A árvore geradora resultante dessa execução é uma MST? Justifique.



QUESTÃO 3 (2,5pt)

Complete o texto a seguir adequadamente.

Um programador recebeu uma série de propostas de trabalho em projetos de curta duração. Cada proposta possui uma dia de início, um dia de fim, e um valor oferecido como pagamento. O objetivo do profissional é escolher que propostas aceitar de forma a maximizar o valor total recebido, sendo que ele deve trabalhar durante todos os dias de cada projeto escolhido

com dedicação exclusiva, ou seja, a cada dia, no máximo um projeto.

Ele pensou então na seguinte solução usando a técnica da _____(a). Primeiramente, ele ordenou os projetos em ordem não decrescente de data de término. Assim, ele passou a ter n projetos na forma (s_i, e_i, p_i), para $i = 1, \ldots, n$, onde s_i corresponde ao dia do ano de início do projeto (numerados $1, 2, 3, \ldots$), $e_i \geq s_i$ corresponde ao dia do final do projeto, e p_i corresponde ao valor oferecido como pagamento. Repare que $e_1 \leq e_2 \leq \cdots \leq e_n$. Ele decidiu então resolver o problema acrescentando progressivamente os projetos, ou seja definindo

S_i = valor total máximo recebível apenas com os primeiros i projetos.

Em particular

S_0 = _____(b).

Para cada projeto $i = 1, \ldots, n$, ele define o 'predecessor' $pred(i)$ como sendo o maior $j < i$ tal que o projeto j não se sobrepõe ao projeto i (por def. $pred(1) = 0$). O programador pode então escolher aceitar ou não o projeto i. Se ele não aceita o projeto i, então

S_i = _____(c),

pois o máximo será determinado apenas pelas escolhas entre os projetos anteriores. Se ele aceita o projeto i, então o valor máximo será dado por

S_i = _____(d),

pois, nesse caso, além do pagamento pelo projeto i, deve-se adicionar o valor máximo que se pode obter com os projetos anteriores que não conflitam com o projeto i. Considerando os dois casos, temos então que, em geral

S_i = _____(e).

Como cada valor de S_i só depende de valores de S_j com $j < i$, então esses valores podem ser calculados do menor para o maior i, sendo os valores armazenados numa tabela. Por exemplo, supondo as propostas $(1, 7, \$5)$, $(3, 10, \$3)$, $(9, 16, \$5)$, $(5, 28, \$4)$, $(19, 32, \$2)$, $(33, 37, \$5)$, e $(13, 40, \$3)$, a tabela seria

$S = \langle$_____$\rangle$(f),

e portanto o valor total máximo seria

$S_{\text{-(g)}}$ = _____(h).

■ QUESTÃO 4 (2,5pt)

Um problema frequentemente encontrado pelos cartógrafos ilustradores é colorir um mapa com a menor quantidade de cores possível de forma que localidades que fazem fronteira não possuam nunca a mesma cor. Nesse problema, o mapa pode ser codificado como um grafo no qual cada localidade é representada por um vértice e uma aresta entre duas localidades indica que elas fazem fronteira. Infelizmente, decidir se um mapa pode ser colorido com 3 cores (R, G, B) é um problema NP-completo. Um algoritmo baaseado em *backtracking* pode ser utilizado para obter uma 3-coloração atribuindo-se progressivamente uma cor a cada vértice do grafo, e retrocedendo assim que a coloração for inválida.

Represente o mapa a seguir como um grafo e ilustre a árvore de execução do algoritmo backtracking para a 3-coloração tendo-o como entrada. Os nós devem ser coloridos na ordem dos rótulos e as cores devem ser escolhidas na ordem R, G, B.

■ ■ ■