

Prova

Questão 1

A solução é observar que o novo fluxo pode aumentar por no máximo um e que podemos tentar encontrar um caminho aumentante no grafo residual. Caso não existe o fluxo já é máximo, caso contrário obtemos o novo fluxo em uma iteração. O grafo residual pode ser construído em tempo $O(m + n)$ e uma busca em largura para encontrar o caminho aumentante também custa somente $O(m + n)$.

Questão 2

Foi esperada uma explicação das operações push (empurrar o excesso atual de um vértice para sucessores de menor altura) e relabel (aumentar a altura de um vértice); uma explicação do algoritmo em geral (inicia com s na altura n , todos restantes na altura 0 , repetidamente escolhe um vértice com excesso e aplica push, ou, caso isso não é possível um relabel).

Questão 3

O laco exterior precisa n iterações, a maioria das operações é constante. Importante são as operações no conjunto S de adicionar um elemento e remover o elemento máximo. Um heap binário implementa as duas operações em $O(\log n)$, logo o algoritmo precisa $O(n \log n)$ tempo total incluindo o tempo para ordenar as tarefas pelo prazo. (Soluções com $O(n \log n)$: 2.5pt, com $o(n^2)$: 1.25, com $\Theta(n^2)$: 0.5pt, com $\omega(n^2)$: 0 pt.

Questão 4

O problema é uma generalização do PCV e para $R = V$ é o PCV. Logo é NP-completa e não possui $p(n)$ -aproximação para qualquer polinômio $p(n)$. Para encontrar uma solução: com força bruta podemos resolver o PCV para todas 2^{n-r} seleções da cidades adicionais para um tempo total de $O(n!2^{n-r})$. Uma modificação simples da programação dinâmica de Bellmann (não visto em aula) consegue resolver o problema em tempo $O(2^r 2^{n-r} \text{poly}(n))$.

(Caso podemos repetir cidades em $V \setminus R$ podemos substituir as distancias entre cidades em R pela distancia mínima entre eles usando vértices em $V \setminus R$ e reduzir o problema para o PCV.)

Questão 5

As complexidades são

Algoritmo	Complexidade	Comentário
Hopcroft-Karp	$\sqrt{nm}$	
Algoritmo Húngaro	$n^2 m$	Como visto em aula
Edmonds-Karp	nm^2	
Push-relabel	n^3	Como vista em aula
Dijkstra	$m \log n$	Com heap binário
Busca em profundidade	m	ou $n + m$

Questão 6

Considerar duas listas consegue balancear as chaves melhor em determinados casos, porém ainda temos, esperadamente α chaves por lista, logo a complexidade da busca continue ser $1 + \alpha$, a mesma do hashing com encadeamento. Similarmente a inserção custo $O(1)$. Logo o colega não tem razão: perdemos a busca em $O(1)$ do cuco hashing, uma operação normalmente mais frequente que a inserção. A abordagem se compara com as outras abordagens similar que o hashing com encadeamento.