Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Sistemas de Informação

IF969 Algoritmos e Estruturas de Dados — 2013.1
Prof. Paulo Fonseca

SEGUNDA PROVA — 11 de Setembro de 2013

- Esta prova contém 04 (quatro) questões.
- A duração da prova é de 1h40min.
- A detecção de cópia implicará na atribuição de nota 0 (zero) à prova.

QUESTÃO 1 (2,5 pts)

Considere o grafo dado pela seguinte lista de adjacências:

$1 \to 2, 4$ $\quad$ $3 \to 4, 5, 6$ $\quad$ $5 \to 4$
$2 \to 5$ $\quad$ $4 \to 2$ $\quad$ $6 \to 1$

Complete o diagrama a seguir correspondente ao percurso em largura a partir do vértice 3,

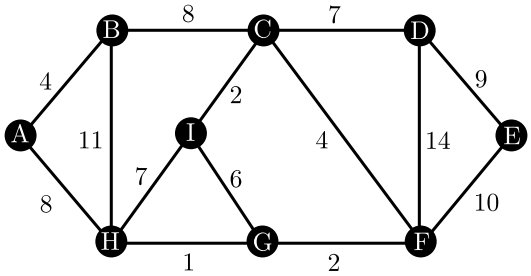
Ordem	P (marcados)	Q (fila)	V (visitados)
1	001000	(3)	( )
⋮	⋮	⋮	⋮

sendo

Ordem: a ordem de execução
P**:** o array booleano dos vértices marcados (enfileirados) para visita
Q**:** a fila de vértices a visitar
V**:** os vértices visitados segundo ordem de visita.

QUESTÃO 2 (2,5 pts)

Considere o grafo



Complete o diagrama a seguir correspondente à execução do Algoritmo Prim a partir do vértice A. Cada célula da tabela tem a forma w/v, onde w representa o peso da aresta a ser escolhida e v o vértice da outra extremidade.

Iteração	A	B	C	D	E	F	G	H	I
1	$\boxed{0/\star}$	$\infty/?$	$\infty/?$	$\infty/?$	$\infty/?$	$\infty/?$	$\infty/?$	$\infty/?$	$\infty/?$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

QUESTÃO 3 (2,5 pts)

Considere o problema da mochila (0/1 Knapsack sem reposição) para a seguinte entrada:

Item	1	2	3	4	5
Peso (w)	4	2	3	5	6
Valor (v)	90	30	80	100	90

Capacidade da mochila: $W = 9$

a) Exiba a tabela de programação dinâmica correspondente à solução dessa instância do problema.
b) Indique quais itens compõem solução ótima, representando na matriz de PD as células percorridas para obter-se essa solução.

QUESTÃO 4 (2,5 pts)

A estrutura de dados de floresta para conjuntos disjuntos constituidos por elementos num universo $\mathcal{A} = \{a_1, \ldots, a_n\}$ pode ser representada por um array $\mathtt{P} = (\mathtt{p}[1], \ldots, \mathtt{p}[n])$, onde $\mathtt{p}[i]$ representa o 'pai' do elemento a_i.

Escreva em pseudo-código uma versão iterativa (não-recursiva) do procedimento *find_pc*, ou seja o *find* com a *heurística compresssão de caminhos*, que recebe um array $\mathtt{P} = (\mathtt{p}[1], \ldots, \mathtt{p}[n])$, como descrito acima, e um valor i, e devolve o representante da classe do elemento a_i. O algoritmo deve ter complexidade $\Theta(n)$.