Projeto e Analise de Algoritmos
Prova 3 – 22/05/2006

Questão 1 [5 pts]

Um ladrão que rouba uma loja encontra n itens: o i-ésimo item vale v[i] reais e pesa w[i] quilos, onde v[i] e w[i] são inteiros. Ele deseja levar uma carga tão valiosa quanto possível, mas pode carregar no máximo W quilos em sua mochila, para algum inteiro W. Que itens ele deve levar?

Obs.: Cada item deve ser levado ou deixado para trás; o ladrão não pode levar uma quantidade fracionária de um item ou levar um item mais de uma vez.

(a) Precisa-se escolher uma ordenação specifica dos itens? Se for o caso, qual seria?
(b) Quais são as subestruturas ótimas do problema? Exiba as duas possiveis subestruturas ótimas que vem de uma destas subestruturas ótimas.
(c) Define recursivamente o valor de c[i, Y], o valor máximo de uma carga escolhida entre os i primeiros itens cujo peso não excede Y.
(d) Forneça um algoritmo de programação dinâmica que seja executado no tempo O(nW) para calcular o valor máximo que o ladrão pode levar assim como uma lista dos itens a ser levado.

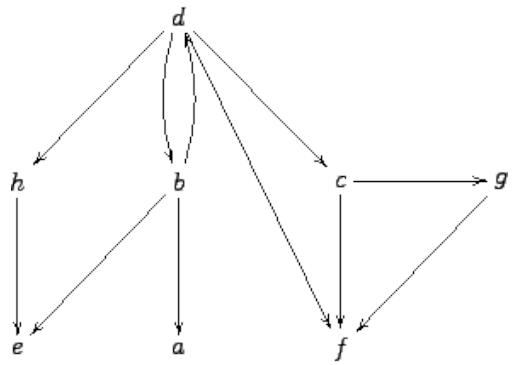
Questão 2 [5 pts]

(a) Modifique o pseudocódigo para a busca em profundidade, de tal modo que ele associa para toda aresta no grafo orientado G o seu tipo: árvore, retorno, diretas ou cruzadas. Pode considerar que as informações sobre as arestas se encontram numa matriz *tipo*, assim consideraremos que o tipo da aresta (u,v) é *tipo*[u,v].

```
DFS(G)
1. para cada vértice u ∈ V[G] faça
2.         cor[u] ← BRANCO
3.         π[u] ← NIL
4. fim-para
5. tempo ← 0
6. para cada vértice u ∈ V[G] faça
7.         se cor[u] = BRANCO
8.                 então DFS-VISTIT(u)
9.         fim-se
10. fim-para
```

```
DFS-VISIT(u)
1. cor[u] ← CINZA
2. tempo ← tempo + 1
3. d[u] ← tempo
4. para cada vértice v ∈ Adj[u] faça
5.         se cor[v] = BRANCO
6.                 então    π[v] ← u
7.                          DFS-VISTIT(v)
8.         fim-se
9. fim-para
10. cor[u] ← PRETO
11. tempo ← tempo + 1
12. f[u] ← tempo
```

(b) Mostre como a busca em profundidade funciona sobre o grafo G abaixo. Suponha que o loop for das linhas 5 a 7 do procedimento DFS considere os vértices em ordem alfabética, e suponha que cada lista de adjacências esteja em ordem alfabética. Mostre os tempos de descoberta e término para cada vértice, e mostre também o tipo de cada aresta.



(c) Qual tipo de busca, visto em aula, podemos utilisar para obter o caminho mais curto entre u e v?