Armando Matos, 2008/2009

Tópicos Avançados em Algoritmos

Programação Dinâmica II e minorantes de complexidade

1. Problema da mochila

   Foi descrito nas aulas teóricas o seguinte algoritmo que usa tabelação ("memoizing") para melhorar a eficiência de um algoritmo para o problema da mochila.

```
Parâmetros: t:  lista dos tamanhos, v: lista dos valores
            i:  próximo elemento a ser ou não seleccionado
            tm: espaço ainda disponível na mochila
Retorna:    valor óptimo
Variável val[i][tm] usada para tabelação; inicializada com NAO_DEF
01 def valor(t,v,i,n,tm):
02   if i>n or tm<=0:
03     return 0
04   if val[i][tm] != NAO_DEF:
05     return val[i][tm]
06   if t[i]>tm:
07     res = valor(t,v,i+1,n,tm)
08   else:
09     res = max(v[i] + valor(t,v,i+1,n,tm-t[i]),valor(t,v,i+1,n,tm))
10   val[i][tm] = res
11   return res
```

   (a) Desenhe a árvore de chamadas (sem usar tabelação, isto é, ignorando as linhas 04, 05 e 10) para `n=4`, `t=[2,1,2,3]`, `v=[1,2,3,1]` e tamanho da mochila `5`. Em cada nó indique o espaço disponível `tm`.

   (b) Mostre que, quando não se usa tabelação, o número de chamadas pode ser 2^n (a eficiência do algoritmo é pois $\Omega(2^n)$).

   (c) Considere agora a situação em que se usa tabelação, ver algoritmo em cima. Mostre que o número de chamadas não excede nT onde T é o tamanho da mochila (valor de `tm` na chamada inicial).

2. Um problema sobre entropia

   Suponha que a variável aleatória x pode assumir 2^n valores, $00\ldots0$, $00\ldots1$,$\ldots 11\ldots1$ (n bits) com probabilidades respectivamente p_1, p_2, p_{2^n}. Quando se verifica a ocorrência de um x_i particular, a informação recebida é $\log(1/p_i)$, tanto maior quanto mais pequena é a probabilidade. À informação média chama-se *entropia* (de Shannon) da distribuição de probabilidade em causa,

$$S = \sum_i p_i \log \frac{1}{p_i}$$

   (a) Como tratar as probabilidades? Mostre que $\lim_{p\to 0} p\log(1/p) = 0$.

(b) Considere as situações seguintes e calcule as correspondentes entropias; interprete os resultados obtidos.

i. Todas as probabilidades são 0, excepto uma (que vale 1).

ii. Todas as probabilidades são 0, excepto duas que valem (ambas) 1/2.

iii. Todas as probabilidades são iguais a $1/2^n$.

3. O "merge-sort" é próximo do óptimo?

Vimos que todo o algoritmo de ordenação (de um vector com n elementos distintos) baseado em comparações, tem que efectuar no pior caso um número de comparações $c(n)$ que satisfaz $2^{c(n)} \geq n!$. Sabemos também que, quando n é uma potência de 2, o número de comparações efectuado pelo "merge-sort" não excede $n \log n - n + 1$. Faça uma tabela que compare este valor com o minorante referido atrás. Que conclui?

4. Caminho hamiltoneano

Considere o problema de, dado um grafo $G = (V, E)$ com um número de vértices conhecido n, determinar um caminho hamiltoneano[1] desse grafo. Usa-se o modelo extermo dos dados com acessos do tipo "$(i, j) \in E$?". Se existir caminho hamiltoneano, o algoritmo escreve esse caminho $(v_1, v_2, \ldots, v_n)$ com a convenção $v_1 < v_n$. Por exemplo, para $n = 3$, pode haver 4 respostas

```
(1,2,3)   (1,3,2)   (2,1,3)   "não existe"
```

(a) Determine em função de n o número $r(n)$ de respostas possíveis.

(b) Mostre que, para cada um dos $r(n) - 1$ ciclos possíveis nas resposta, existem grafos que têm unicamente esse ciclo hamiltoneano. Por exemplo, para $n = 3$ há um grafo que tem unicamente o ciclo `(1,2,3)`, um grafo que tem unicamente o ciclo `(1,3,2)`, etc.

(c) Usando o princípio da informação necessária determine um minorante do número de acessos do tipo indicado.

[1] Um caminho que passa uma e uma só vez por cada vértice do grafo e em que os vértices de partida e de chegada são distintos (não é um ciclo).