Opcional Segundo examen
Fundamentos de Análisis y Diseño de algoritmos
Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle

Prof. Juan Francisco Díaz F.

17 de Diciembre de 2007

Nombre:________________________________
Código:________________

El problema de la Repartición Óptima de Cupos (ROC)

Un motivo para una frecuente, y aparentemente justa, insatisfacción estudiantil se presenta cada semestre entre los estudiantes que se matriculan al final, pues no encuentran cupos para las materias que quieren ver. Cada semestre aparecen nuevas estrategias para resolver las injusticias que se descubren de proceso en proceso: matricular por código de los antiguos a los más nuevos, matricular por código de los nuevos a los más antiguos, matricular por promedio general, ...; sin embargo siempre aparecen los insatisfechos, explicando con buenas razones por qué fue injusta la estrategia, y proponiendo otra manera más adecuada de repartir los cupos.

Se desea entonces encotrar una manera *óptima* de repartir los cupos, es decir, una manera que genere un bajo grado de insatisfacción entre los estudiantes.

Formalmente el problema de *la repartición óptima de cupos*, ROC, se describe a continuación. Sean:

- $P = \{1, 2, 3, 4, 5\}$, el conjunto de prioridades de materias disponibles.
- $M = \{(M_i, m_i) : i = 1 \ldots k\}$ el conjunto de materias disponibles, donde M_i es el código de la materia i, y m_i es cupo de estudiantes para M_i.
- $E = \{(e_i, ms_i) : i = 1 \ldots r\}$, donde e_i es el código del estudiante i, y $ms_i = \{(s_{ij}, p_{ij}) : 1 \leq j \leq 7\}$, es el conjunto de materias solicitadas por el estudiante i y priorizadas por él (s_{ij} es el código de la j-ésima materia solicitada por el estudiante i y $p_{ij} \in P$ es la prioridad de s_{ij}).

  Además, deben cumplirse las siguientes restricciones:

  - $\sum_{l=1}^{|ms_i|} p_{il} \leq \gamma(|ms_i|)$ para $i = 1 \ldots r$, donde $\gamma(X) = 3X - 1$.
  - para todo $i = 1 \ldots r$, $s_{ij} \neq s_{ik}$ si $j \neq k$.
- $A = \{(e_i, ma_i) : i = 1 \ldots r\}$ el conjunto de materias asignadas por estudiante, donde $ma_i \subseteq ms_i$. Además A no debe sobrepasar el cupo en ninguna materia.
-
  - $f_i = (1 - \frac{|ma_i|}{|ms_i|}).(\frac{\sum_j \{p_{ij} : (s_{ij}, p_{ij}) \notin ma_i\}}{\gamma(|ms_i|)})$, la función de medición de insatisfacción del estudiante i.
  - $\mathcal{F}_{\langle M,E \rangle}(A) = \frac{\sum_{i=0}^{r} f_i}{r}$, la función de insatisfacción general de los estudiantes, dados una entrada $\langle M, E \rangle$ y una solución A.

El problema ROC

- **Entrada:** La tupla $\langle M, E\rangle$
- **Salida:** Un conjunto A tal que $\mathcal{F}_{\langle M,E\rangle}(A)$ sea mínima.

1. **Calentamiento [15 pts.]**

   Suponga que se cuenta con una función $Pr(i,j)$, que calcula la prioridad con que el estudiante e_i solicitó la materia $M_j, 1 \leq i \leq r, 1 \leq j \leq k$. Si e_i no solicitó $M_j, Pr(i,j) = 0$.

   Para cada materia $M_j, 1 \leq j \leq l$, se define S_j como el conjunto de estudiantes que solicitaron la materia M_j :
   $$S_j = \{e_i : Pr(i,j) \neq 0\}.$$

   Dado $S \subseteq S_j$, se define la insatisfacción generada por S sobre la materia M_j como:
   $$In(S, S_j) = 1 - \frac{\sum_{e_i \in S} Pr(i,j)}{\sum_{e_i \in S_j} Pr(i,j)}.$$

   Considere el siguiente algoritmo para resolver el problema ROC:

   $ROC\_Ingenuo(M,E)$
   1 **for** $i = 1$ **to** r
   2   **do** $ma_i \leftarrow \emptyset$
   3 **for** $j = 1$ **to** k
   4   **do** Calcule S_j
   5   **if** $|S_j| \leq m_j$
   6   **then** $A \leftarrow S_j$
   7   **else** $I \leftarrow 1, A \leftarrow \emptyset$
   8   **foreach** $S \subseteq S_j : |S| = m_j$
   9   **do** $I_S \leftarrow In(S, S_j)$
   10   **if** $I_S < I$
   11   **then** $I \leftarrow I_S, A \leftarrow S$
   12   **foreach** $i \in A$
   13   **do** $ma_i \leftarrow ma_i \cup \{(M_j, Pr(i,j)\}$
   14 **return** $\{(e_i, ma_i), i = 1..r\}$

   a) **[5 pts.]** Suponga que la entrada es:
   - $M = \{(M_1, 3), (M_2, 4), (M_3, 2)\}$
   - $E = \{(e_1, ms_1), (e_2, ms_2), (e_3, ms_3), (e_4, ms_4), (e_5, ms_5)\}$, donde:
     - $ms_1 = \{(M_1, 5), (M_2, 2), (M_3, 1)\}$
     - $ms_2 = \{(M_1, 4), (M_2, 1), (M_3, 3)\}$
     - $ms_3 = \{(M_2, 3), (M_3, 2)\}$
     - $ms_4 = \{(M_1, 2), (M_3, 3)\}$
     - $ms_5 = \{(M_1, 3), (M_2, 2), (M_3, 3)\}$

   Qué devuelve como solución $ROC\_Ingenuo(M,E)$? Cuál es el valor de la insatisfacción de esa solución?

   b) **[10 pts.]** Describa una fórmula para calcular el orden de complejidad del algoritmo en función de k, r, m_j $j = 1..k$, y $|S_j| j = 1..k$.

2. **Algoritmos voraces [20 pts.]**

   Sea $S_j^{ord}[1..|S_j|]$ un arreglo que representa el conjunto S_j ordenado descendentemente por prioridad. Considere la siguiente modificación al algoritmo $ROC\_Ingenuo$ (las líneas 6-11 son reemplazadas por las líneas 6-7):

   $ROC\_Voraz(M,E)$
   1 **for** $i = 1$ **to** r
   2   **do** $ma_i \leftarrow \emptyset$
   3 **for** $j = 1$ **to** k
   4   **do** Calcule S_j^{ord}
   5   **if** $|S_j| \leq m_j$
   6   **then** $A \leftarrow S_j^{ord}[1..|S_j|]$
   7   **else** $A \leftarrow S_j^{ord}[1..m_j]$
   8   **foreach** $i \in A$
   9   **do** $ma_i \leftarrow ma_i \cup \{(M_j, Pr(i,j)\}$
   10 **return** $\{(e_i, ma_i), i = 1..r\}$

   a) **[5 pts.]** Pronúnciese brevemente sobre la siguiente afirmación:
   $$ROC\_Ingenuo(M,E) = ROC\_Voraz(M,E),$$
   para todo M, E.

b) **[15 pts.]** ROC_Voraz encuentra siempre el óptimo? Justifique su respuesta.

3. **Programación Dinámica [65 pts.]**

Analizando el problema para intentar solucionarlo usando Programación Dinámica un grupo de estudiantes hizo la siguiente afirmación:

> Sea $A_1 = \{(e_i, ma_i) : i = 1..r\}$ una solución al problema cuya entrada es $\langle\{(M_j, m_j) : j = 1..k\},\ \{(e_i, ms_i) : i = 1..r\}\rangle$. Si A_1 es óptima entonces $A_2 = \{(e_i, ma_i) : i = 2..r\}$ es una solución óptima para el problema cuya entrada es $\langle\{(M_j, c_j) : j = 1..k\},\ \{(e_i, ms_i) : i = 2..r\}\rangle$, donde
>
> $$c_j = \begin{cases} m_j - 1 & \text{si } (M_j, \_) \in ma_1 \\ m_j & \text{sino} \end{cases}$$

a) **Caracterizar la estructura de una solución óptima. [20 pts.]**

1) **[10 pts.]** La afirmación anterior es verdadera o falsa? Justifique su respuesta (probándola o dando un contraejemplo).

2) **[10 pts.]** Suponga que la afirmación es verdadera. A partir de ella, se pueden definir los problemas $\Pi_{i,(c_j)_{j=1..k}}, 1 \leq i \leq r, 0 \leq c_j \leq m_j$, para $j = 1..k$ así:

Problema $\Pi_{i,(c_j)_{j=1..k}}$

- **Entrada:** $\langle\{(M_j, c_j) : j \in 1..k, 0 \leq c_j \leq m_j\},$ $\{(e_h, ms_h) : h \in i..r\}\rangle$
- **Salida:** Un conjunto $A = \{(e_h, ma_h) : h = i..r\}$ tal que $\mathcal{F}(A)$ sea mínima.

a' Cuál sería el problema original? ________________

b' Cuantos subproblemas hay? ________________

b) **Defina recursivamente el valor de una solución óptima. [10 pts.]**

Sea $\mathcal{F}[i, \vec{c}]$ el valor óptimo de una solución al problema $\Pi_{i,(c_j)_{j=1..k}}$. Por otro lado, dados i y $S \subseteq ms_i$ sea $Ins(i, S)$ la insatisfacción del estudiante e_i si el conjunto de materias asignadas a él fuera S. Es decir,

$$Ins(i, S) = \frac{|ms_i - S|}{|ms_i|} \frac{\sum_{(M_t,p)\in(ms_i - S)} p}{\gamma(|ms_i|)}.$$

Defina recursivamente $\mathcal{F}[i, \vec{c}]$ para todo $1 \leq i \leq r, 0 \leq c[j] \leq m_j, j = 1..k$.

$$\mathcal{F}[i, \vec{c}] = \begin{cases} & \text{Si} \ldots \\ \vdots & \vdots \\ & \text{Si} \ldots \end{cases}$$

c) **Calcule el valor de una solución óptima. [15 pts.]**

Como resultado de este punto se espera un algoritmo que calcule el costo de una solución óptima al problema original.

d) **Construya una solución óptima. [10 pts.]**

Como resultado de este punto se espera:

- Incluir en algún punto del algoritmo anterior, si es necesario, una estructura que le permita construir una solución óptima.
- Un algoritmo que recorra esa estructura construyendo (o imprimiendo) una solución de costo óptimo.

e) **Complejidad [10 pts.]** Describa una fórmula para calcular el orden de complejidad en espacio y tiempo del algoritmo en función de k, r y m_j $j = 1..k$.

OJO: Ud. puede pedir una hoja de ayuda para el problema 3.b a un costo de 10 pts.

Ayuda para el problema 3.b [-10 pts.]

Dados $i, \vec{c}$, sea $S = \{(M_t, p) \in ms_i : c[t] \neq 0\}$; o sea S es el conjunto de materias que ha solicitado el estudiante e_i en las que queda cupo según la oferta $\vec{c}$.

$$\mathcal{F}[i, \vec{c}] = \begin{cases} Ins(i, S) & \text{Si } i = r \\ Min_{B \subseteq S} \frac{\{Ins(i,B)+(r-i)\mathcal{F}[i+1, \vec{c_B}]\}}{r-i+1} & \text{Si } i < r \end{cases},$$

donde

$$c_B[j] = \begin{cases} c[j] - 1 & \text{si } (M_j, \_) \in B \\ c[j] & \text{sino} \end{cases}.$$

Ayuda para el problema 3.b [-10 pts.]

Dados $i, \vec{c}$, sea $S = \{(M_t, p) \in ms_i : c[t] \neq 0\}$; o sea S es el conjunto de materias que ha solicitado el estudiante e_i en las que queda cupo según la oferta $\vec{c}$.

$$\mathcal{F}[i, \vec{c}] = \begin{cases} Ins(i, S) & \text{Si } i = r \\ Min_{B \subseteq S} \frac{\{Ins(i,B)+(r-i)\mathcal{F}[i+1, \vec{c_B}]\}}{r-i+1} & \text{Si } i < r \end{cases},$$

donde

$$c_B[j] = \begin{cases} c[j] - 1 & \text{si } (M_j, \_) \in B \\ c[j] & \text{sino} \end{cases}.$$

Ayuda para el problema 3.b [-10 pts.]

Dados $i, \vec{c}$, sea $S = \{(M_t, p) \in ms_i : c[t] \neq 0\}$; o sea S es el conjunto de materias que ha solicitado el estudiante e_i en las que queda cupo según la oferta $\vec{c}$.

$$\mathcal{F}[i, \vec{c}] = \begin{cases} Ins(i, S) & \text{Si } i = r \\ Min_{B \subseteq S} \frac{\{Ins(i,B)+(r-i)\mathcal{F}[i+1, \vec{c_B}]\}}{r-i+1} & \text{Si } i < r \end{cases},$$

donde

$$c_B[j] = \begin{cases} c[j] - 1 & \text{si } (M_j, \_) \in B \\ c[j] & \text{sino} \end{cases}.$$

Ayuda para el problema 3.b [-10 pts.]

Dados $i, \vec{c}$, sea $S = \{(M_t, p) \in ms_i : c[t] \neq 0\}$; o sea S es el conjunto de materias que ha solicitado el estudiante e_i en las que queda cupo según la oferta $\vec{c}$.

$$\mathcal{F}[i, \vec{c}] = \begin{cases} Ins(i, S) & \text{Si } i = r \\ Min_{B \subseteq S} \frac{\{Ins(i,B)+(r-i)\mathcal{F}[i+1, \vec{c_B}]\}}{r-i+1} & \text{Si } i < r \end{cases},$$

donde

$$c_B[j] = \begin{cases} c[j] - 1 & \text{si } (M_j, \_) \in B \\ c[j] & \text{sino} \end{cases}.$$