PAC3

Presentació

Aquesta PAC aprofundeix en el concepte de complexitat computacional que cobreix els continguts estudiats en els mòduls 6 i 7 de l'assignatura. Els exercicis treballen els conceptes de mesures de complexitat, la reducció i completesa, la classe NP-complet i alguns dels problemes intractables més importants que es coneixen.

Competències

En aquesta PAC es treballen les següents competències del Grau en Enginyeria Informàtica:

- Capacitat per utilitzar els fonaments matemàtics, estadístics i físics per comprendre els sistemes TIC.
- Capacitat per analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per a resoldre'l.

Objectius

Els objectius concrets d'aquesta PAC són:

- Entendre els conceptes d'intractabilitat i no-determinisme.
- Conèixer les diferents classes de complexitat i saber classificar els problemes en cada una d'aquestes.
- Entendre el concepte de reducció entre problemes i saber demostrar quan un problema és NP-complet.
- Reconèixer problemes intractables que apareixen de forma habitual en informàtica i en enginyeria.
- Entendre i saber aplicar les tècniques bàsiques de reducció polinómica dels problemes NP-complets.

Descripció de la PAC

1. (Valoració d'un 20%) Definiu cadascun dels problemes següents en forma de funció, i classifiqueu-los com a problemes de decisió, càlcul o optimització. Reformuleu els de càlcul i els d'optimització com a problemes de decisió i indiqueu a quina classe de complexitat pertanyen:

   (a) Donat un graf $G = (V, A)$ i dos vèrtexs $u, v \in V$, determinar si existeix un camí de u a v.

   (b) Donat un graf $G = (V, A)$ connex, obtenir un arbre generador de G amb el nombre màxim de fulles (vèrtexs de grau 1).

   (c) Donada una llista d'enters ordenada L de longitud n i un enter e, trobar la posició que ocupa e a L.

   (d) Donat un graf $G = (V, A)$, obtenir una partició dels seus vèrtexs, (V_1, V_2), de manera que $TALL(V_1, V_2) = |\{(u, v) \in A \mid u \in V_1, v \in V_2\}|$ sigui màxim.

Solució:

(a) ACCG : $\mathcal{G} \to \{\text{SÍ}, \text{NO}\}$. És un problema de decisió. Els algorismes BFS i DFS el resolen en temps $O(n+m)$, on n és l'ordre del graf i m és la mida del graf. Com que la mida és, com a molt, quadràtica respecte a l'ordre n del graf, l'algorisme és $O(n^2)$ i el problema és a la classe P.

(b) ARBRE-GEN-MAX-FULLES : $\mathcal{G} \to \mathcal{P}(A)$. És un problema d'optimització. La funció de valoració és la suma del nombre de fulles de l'arbre generador. La versió de decisió seria, *donat un graf connex $G = (V, A)$ i un valor k, decidir si existeix un arbre generador de G amb un nombre de fulles $\geq k$*. El problema és NP ja que si ens donen un possible arbre generador com a testimoni podem comprovar que ho és i que té un nombre de fulles igual o superior a k en temps polinòmic.

(c) CERCA-DICOT : $\mathbb{Z}^* \times \mathbb{Z} \to \mathbb{Z}$. És un problema de càlcul. Com que el vector està ordenat podem fer una cerca dicotòmica que és $O(\log n)$ i per tant el problema és a la classe P. La versió de decisió seria, *donada una llista d'enters ordenada L i un enter e, decidir si e apareix en la llista.*

(d) TALL-MAX : $\mathcal{G} \to (V \to \{1, 2\})$. És un problema d'optimització. La funció de valoració és $TALL(V_1, V_2)$, el nombre d'arestes existents entre els vèrtexs de V_1 i V_2. La versió de decisió seria, *donat un graf $G = (V, A)$ i un valor k, decidir si existeix una partició dels seus vèrtexs tal que $tall(V_1, V_2) \geq k$.* El problema és NP ja que si ens donen una possible partició com a testimoni podem comprovar que és correcta i que el nombre d'arestes entre els vèrtexs de V_1 i V_2 és igual o superior a k en temps polinòmic.

---

2. (Valoració d'un 15%) Considereu les fórmules següents,

   - $(a \wedge \bar{b}) \vee c$.
   - $(a \vee \bar{b}) \wedge ((\bar{b} \vee c) \wedge a)$.
   - $(a \vee b) \wedge (a \vee \bar{c}) \wedge (b \vee c)$.
   - $a \vee (b \wedge c) \vee (\bar{b} \wedge c)$.

   (a) Trobeu totes les assignacions que satisfan cadascuna.

   (b) Indiqueu quines estan en FNC i quines no. Passeu a FNC les que no ho estan.

   (c) Quines fórmules són possibles entrades del problema SAT? I, un cop passades totes a FNC, quines ho són del problema 3SAT?

Solució:

(a) La primera es verifica per $a = 0, b = 0, c = 1$; per $a = 0, b = 1, c = 1$; per $a = 1, b = 0, c = 0$; per $a = 1, b = 0, c = 1$; i per $a = 1, b = 1, c = 1$.
La segona per $a = 1, b = 0, c = 0$; per $a = 1, b = 0, c = 1$; i per $a = 1, b = 1, c = 1$.
La tercera per $a = 0, b = 1, c = 0$; per $a = 1, b = 0, c = 1$; per $a = 1, b = 1, c = 0$ i per $a = 1, b = 1, c = 1$.
L'última per $a = 0, b = 0, c = 1$; per $a = 0, b = 1, c = 1$; per $a = 1, b = 0, c = 0$; per $a = 1, b = 0, c = 1$; per $a = 1, b = 1, c = 0$; i per $a = 1, b = 1, c = 1$.

(b) La segona i la tercera estan en FNC. La primera seria: $(a \vee c) \wedge (\bar{b} \vee c)$. La tercera quedaria: $a \vee c$.

(c) En la forma original només la segona i la tercera poden ser entrada de SAT, i un cop passades a FNC ho són totes. Cap ho és de 3SAT.

---

3. (Valoració d'un 30%) Volem trobar un arbre generador d'un graf connex en que cap vèrtex tingui un grau major que un cert nombre k. Anomenem al problema ARBGENGAF (arbre generador de grau afitat).

(a) Descriviu de manera formal el problema ARBGENGAF.

(b) El problema ARBGENGAF és un problema de decisió, de càlcul o d'optimització? Descriviu de manera formal els altres dos problemes associats.

(c) Demostreu que el problema ARBGENGAF és verificable en temps polinòmic.

(d) Expliqueu quines semblances hi ha entre el problema decisional ARBGENGAFDEC corresponent al problema ARBGENGAF i el problema decisional HAM_CYCLE que consisteix en determinar si existeix un cicle hamiltonià en un graf.

(e) Demostreu que HAM_CYCLE $\leq_p$ ARBGENGAFDEC.

(f) A partir dels resultats obtinguts en els apartats anteriors, a quina classe de complexitat podem afirmar que pertany el problema ARBGENGAFDEC?

Solució:

(a) Donat un graf $G = (V, A)$ connex i un nombre natural k, trobar un graf $G' = (V, A')$ connex tal que $A' \subseteq A$, $|A'| = |V| - 1$ i $\forall v \in V \ |\{(u, v) \in A' \mid u \in V\}| \leq k$.

(b) El problema ARBGENGAF és un problema de càlcul.
El problema de decisió associat ARBGENGAFDEC és: Donat un graf $G = (V, A)$ connex i un nombre natural k, existeix un graf $G' = (V, A')$ connex tal que $A' \subseteq A$, $|A'| = |V| - 1$ i $\forall v \in V \ |\{(u, v) \in A' \mid u \in V\}| \leq k$? El problema d'optimització associat ARBGENGAFOPT és: Donat un graf $G = (V, A)$ connex, trobar un graf $G' = (V, A')$ connex tal que $A' \subseteq A$, $|A'| = |V| - 1$ de manera que $\forall v \in V \ |\{(u, v) \in A' \mid u \in V\}|$ sigui mínim.

(c) Un testimoni seria un arbre generador de G en que cap vèrtex té un grau major que k, és a dir, un graf $G' = (V, A')$ connex tal que $A' \subseteq A$, $|A'| = |V| - 1$ i $\forall v \in V \ |\{(u, v) \in A' \mid u \in V\}| \leq k$. La comprovació d'aquestes condicions es pot fer en temps polinòmic.

(d) Si ens restringim a les instàncies del problema decisional ARBGENGAFDEC en les quals $k = 2$ (hi ha un arbre generador en què cap vèrtex tingui un grau major que 2?), el grau 2 correspon a l'arribada i la sortida de cada vèrtex en el cicle hamiltonià (recordeu que es tracta de recòrrer tots els vèrtexs d'un graf sense repetir-ne cap i, per tant, en cada vèrtex només incideixen dues arestes, la d'arribada i la de sortida). Si agafem n vèrtexs i els ajuntem amb $n - 1$ arestes, amb la restricció que cada vèrtex pot aparèixer com a màxim en dues arestes, observem que ens queden tots els vèrtexs amb grau 2 excepte dos amb grau 1. Si connectem els dos vèrtexs amb grau 1, obtenim un cicle hamiltonià (un recorregut que visita tots els vèrtexs un sol cop partint d'un d'aquests i tornant al mateix).

(e) En l'apartat anterior hem vist que si hi ha un cicle hamiltonià en G és que hi ha un arbre generador amb grau no superior a 2. Per tant, HAM_CYCLE $\leq_p$ ARBGENGAFDEC.

(f) En primer lloc, sabem que el problema decisional HAM_CYCLE és un problema NP-complet (veure pàgina 26 del mòdul 7). Per l'apartat (c) sabem que el problema ARBGENGAF pertany a NP (i el problema ARBGENDEC també). I per l'apartat (e) sabem que HAM_CYCLE $\leq_p$ ARBGENGAFDEC. Tot plegat, ens porta a afirmar que el problema ARBGENGAFDEC és també NP-complet.

---

4. (Valoració d'un 20%) Digueu si són certes o falses les afirmacions següents, justificant la resposta:

   (a) El problema "Determinar si existeix un subgraf complet d'un graf G d'ordre k" pertany a NP perquè si ens donen un subgraf complet de k vèrtexs del graf G tenim un testimoni.

   (b) Si $A \leq_p B$ i $B \leq_p A$, aleshores A i B són NP-complets.

   (c) Si A és NP-difícil segur que no és NP-complet.

   (d) Si es demostrés que un determinat problema $A \notin P$ i $A \in NP$ però no és NP-complet, aleshores s'hauria demostrat que $P = NP$.

Solució:

(a) Certa. Donat un subconjunt de k vèrtexs de G podem comprovar en temps polinòmic ($O(k^2)$) que és subgraf complet de G.

(b) Falsa. També poden ser els dos a P. Els dos estaran en la mateixa classe, però no podem afirmar quina és.

(c) Falsa. Si a més de ser NP-difícil $A \in NP$, aleshores A és NP-complet (com per exemple passa amb el problema SAT).

(d) Falsa. S'hauria demostrat el contrari, que $P \neq NP$, perquè voldria dir que hi ha problemes en NP que no són NP-complets.

---

5. (Valoració d'un 15%) Considereu el problema de la motxilla.

   (a) Quina és la solució del problema si $C = 30$, $U = \{1, 2, 3, 4\}$ amb pesos $w = \{17, 11, 14, 7\}$ i valors $v = \{34, 33, 56, 35\}$?

   (b) Sabem que el problema de la motxilla és intractable i per tant, és interessant trobar algorismes que donin una solució aproximada en temps polinòmic. Us donem una heurística voraç perquè la proveu amb l'exemple donat: ordenar els objectes per valor/pes (de major a menor) i anar-los escollint mentre no superem la capacitat total C (els objectes que ens portin a superar-la els rebutgem). Quina és la solució? En cas que retorni l'òptim, doneu un exemple en que no sigui així.

   (c) Què haurem de canviar en l'enunciat del problema perquè l'algorisme de l'apartat anterior ens retorni sempre l'òptim? Justifiqueu la resposta.

Solució:

(a) La taula següent ens mostra totes les possibilitats per a cercar la solució del problema:

Subconjunt	Pes total	Valor total
$\emptyset$	0	0
$\{1\}$	17	34
$\{2\}$	11	33
$\{3\}$	14	56
$\{4\}$	7	35
$\{1,2\}$	28	67
$\{1,3\}$	31	no factible
$\{1,4\}$	24	69
$\{2,3\}$	25	89
$\{2,4\}$	18	68
$\mathbf{\{3,4\}}$	**21**	**91**
$\{1,2,3\}$	42	no factible
$\{1,2,4\}$	35	no factible
$\{1,3,4\}$	38	no factible
$\{2,3,4\}$	32	no factible
$\{1,2,3,4\}$	49	no factible

La solució està en negreta: la millor opció és triar els objectes 3 i 4 amb un pes total de 21 i un valor total de 91.

(b) Ordenats per valor/pes els objectes són: $\{4,3,2,1\}$ (ja que valor/pes $= \{5,4,3,2\}$). Agafaríem en primer lloc l'objecte 4 amb pes total igual a 7 i valor total igual a 35, en segon lloc l'objecte 3 amb pes total igual a 21 i valor total igual a 91, i els altres dos objectes es rebutjarien perquè ens fan superar la capacitat total (32 i 38, respectivament). Per tant, ens dóna la mateixa solució.

Un contraexemple que no ens retorna l'òptim seria: $C = 30$, $U = \{1,2,3,4\}$ amb pesos $w = \{12,11,14,7\}$ i valors $v = \{24,33,56,35\}$. La taula següent ens mostra totes les possibilitats per a cercar la solució del problema:

Subconjunt	Pes total	Valor total
$\emptyset$	0	0
$\{1\}$	12	24
$\{2\}$	11	33
$\{3\}$	14	56
$\{4\}$	7	35
$\{1,2\}$	23	57
$\{1,3\}$	26	80
$\{1,4\}$	19	59
$\{2,3\}$	25	89
$\{2,4\}$	18	68
$\{3,4\}$	21	91
$\{1,2,3\}$	37	no factible
$\mathbf{\{1,2,4\}}$	**30**	**92**
$\{1,3,4\}$	33	no factible
$\{2,3,4\}$	32	no factible
$\{1,2,3,4\}$	44	no factible

La solució està en negreta: la millor opció és triar els objectes 1, 2 i 4 amb un pes total de 30 i un valor total de 92.

Ordenats per valor/pes els objectes són: $\{4,3,2,1\}$ (ja que valor/pes $= \{5,4,3,2\}$). Agafaríem en primer lloc l'objecte 4 amb pes total igual a 7 i valor total igual a 35, en segon lloc l'objecte 3 amb pes total igual a 21 i valor total igual a 91, i els altres dos objectes es rebutjarien perquè ens fan superar la capacitat total (32 i 33, respectivament). Per tant, no ens dóna la mateixa solució.

L'heurística decideix agafar un objecte si encara hi cap (el 3) encara que deixi molt espai buit ($30-21 = 9$) i això impedeix considerar altres opcions amb objectes amb menor valor/pes (1 i 2) que junts ocupen més espai (23 en front de 14) i proporcionen més valor total (57 en front de 56).

(c) Si en lloc de considerar objectes sencers (l'agafo tot o res) permetéssim agafar-ne un fragment o fracció, el nostre algorisme voraç ens retornaria sempre l'òptim. Del primer objecte que no hi cap sencer n'agafem exactament la part que hi cap (i per tant, sempre ocuparem tota la capacitat C).

Al triar els objectes de major a menor valor per unitat de pes estem aprofitant de la millor manera l'espai disponible, i com que l'ocupem en la seva totalitat no hi ha alternatives millors.

Recursos

Recursos Bàsics

- Mòdul didàctic 6. Complexitat computacional.
- Mòdul didàctic 7. Problemes intractables.
- Col·lecció de problemes

Recursos Complementaris

- PACs i exàmens de semestres anteriors.
- Programari per a l'estudi d'algorismes sobre grafs.
- Enllaç: Applets interactius sobre algorismes de grafs.

Criteris d'avaluació

- La PAC s'ha de resoldre **de forma individual**.
- És necessari justificar la resposta a cadascun dels apartats. Es valorarà tant la correctesa de la resposta com la justificació donada.
- En els apartats on calgui aplicar algun algorisme, es valorarà la tria de l'algorisme apropiat, els passos intermedis, el resultat final i les conclusions que se'n derivin.

Format i data de lliurament

Cal lliurar **un únic document** PDF amb les respostes a tots els exercicis. El nom del fitxer ha ser: **PAC3_Cognom1Cognom2Nom.pdf**.

Aquest document s'ha de lliurar a l'espai de **Lliurament i Registre d'AC** de l'aula **abans de les 23:59 del dia 18/12/2013**. **No s'acceptaran lliuraments fora de termini.**