Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona

Cognoms, Nom	D.N.I.

Titulació: EI/ETIG
Assignatura: Anàlisi i Disseny d'Algorismes
Duració: 2 hores
Curs: Q1 2007-2008 (2on Parcial)
Data: 10 de Gener de 2008

1. **(2.5 punts)** La regió d'Aicenev, al sud del Balquistan, és famosa en el món sencer pel seu sistema de comunicacions basat en canals (grans i petits) que es ramifiquen. Tots els canals són navegables i totes les localitats de la regió se situen al costat d'algún canal. Un cop a l'any els alcaldes de la regió es reuneixen en la localitat de San Marco i cadascú d'ells viatja en el seu vaporetto oficial. Proposeu un algorisme als alcaldes que els permeti trobar la manera d'arribar a San Marco de manera que entre tots ells s'hagi fet el mínim recorregut de distància.

   A conseqüència de l'escalfament global, els alcaldes han decidit compartir els vaporettos. Cadascú va en el seu fins a alguna altra ciutat, on es pot unir a un altre o altres alcades, que aniran plegats fins algún altre lloc, formant així grups que es dirigeixen fins a San Marco. Serveix la solució proposada abans? Per què? Si no serveix, proposeu una solució a aquesta nova situació, que minimitzi la distància recorreguda pels vaporettos (no necessàriament pels alcaldes).

   SOLUCIÓ:

Cognoms, Nom | D.N.I.

2. **(1 punt)** El problema Subgraf-Esborrat consisteix en, donats dos grafs G_1 i G_2, determinar si un graf isomorf a G_1 s'obté a partir de G_2 tot esborrant arestes de G_2.

   Demostreu que Subgraf-Esborrat és un problema NP. A continuació demostreu que és NP-complet, fent servir una reducció del problema NP-complet Graf-Hamiltonià a Subgraf-Esborrat.

SOLUCIÓ:

Cognoms, Nom D.N.I.

3. **(3.5 punts)** Suposeu que sou els responsables d'una empresa dedicada a tallar barres d'acer. Donada una barra de M metres, s'obtenen $n+1$ barres més curtes tallant la barra original en n punts, $0 < p_1 < p_2 < \cdots < p_n < M$, on M i els p_i's són valors enters. Podem pensar que els punts de tall p_i ja están marcats a sobre de la barra original.

   Cada tall costa un euro per cada metre que tingui la barra abans de tallar-la. Dissenyeu i analitzeu un algorisme de programació dinàmica que calculi el cost mínim de tallar una barra, donats M, n i $p_1, p_2, \ldots, p_n$; l'algorisme ha de calcular l'ordre òptim dels talls, doncs els punts de tall ja s'han fixat. Escriviu i expliqueu la recurrència corresponent. Escriviu llavors un algorisme iteratiu de programació dinàmica que calculi els costos, basant-se en la recurrència prèviament obtinguda, i analitzeu el seu cost en funció de n i M.

   Per exemple, si $M = 9$, $n = 2$, $p_1 = 2$ i $p_2 = 6$, tallant primer a p_1 el cost és $9+7 = 16$, mentre que tallant primer a p_2 el cost només és $9 + 6 = 15$.

SOLUCIÓ:

4. **(3 punts)** En un tauler amb $n \times n$ escacs es col·loca un cavall en una casella donada. Dissenyeu un algorisme per trobar si existeix alguna forma d'aplicar $n^2 - 1$ moviments de cavall de forma que el cavall visiti totes les caselles del tauler.

   Per exemple, aquesta és una possible solució per a un tauler 5×5 començant des del centre:

22	5	16	11	24
15	10	23	6	1
4	21	♞	17	12
9	14	19	2	7
20	3	8	13	18

   Completeu la plantilla donada més abaix, substituint cada bloc `<<X>>` pel que toqui o especifiqueu que no cal posar codi en el lloc corresponent. Cada bloc pot consistir en una sola línia de codi o més d'una. En algún cas trobareu una pista al costat sobre el tros de codi que cal escriure.

   Es tindrà en compte la claredat d'explicació que hi poseu, l'ús de marcatges i la qualitat general de la vostra solució. La correcció de l'algorisme no garanteix per si sola la nota màxima.

SOLUCIÓ:

Explicacions addicionals sobre la plantilla:

Donada una casella (i, j), $0 \leq i < n$, $0 \leq j < n$ la numerarem amb $k = i \cdot n + j$. Així la casella $(0, 0)$ tindrà el número 0, la $(0, 1)$ el número 1, ..., la $(0, n-1)$ serà la $n-1$, la $(1, 0)$ serà la n, etc. Les conversions entre posicions i els números les fan els mètodes privats `pos2num` i `num2pos` donats més avall. Si una posició p no és vàlida llavors `pos2num`$(p) = -1$. No cal que implementeu aquests mètodes.

El mètode `salt` rep una posició i un valor d, $1 \leq d \leq 8$ i retorna la posició corresponent: el salt $d = 1$ consisteix en anar dos caselles amunt i una la dreta, el salt $d = 2$ dos caselles amunt i una a l'esquerra, i així successivament, en l'ordre contrari al de les agulles del rellotge.

El vector `solucio_` ens dóna els números de les successives caselles per ón s'han de fer els salts del cavall. En l'exemple dibuixat a l'enunciat tindríem `solucio_` $= [12, 9, 18, 21, 10, 1, \ldots]$. Fixeu-vos que, en canvi, el dibuix mostra, per a cada casella, en quin moment passa el cavall per ella.

```
class SaltCavall {
    struct posicio {
       int fil, col;
       posicio(int f, int c) : fil(f), col(c) {};
    };

    int pos2num(posicio p) const;
    posicio num2pos(int r) const;
    posicio salt(posicio p, int d) const;

    vector<int> solucio_;
    bool hihasol_;
    int n_;

    <<A>>

public:
    SaltCavall(int n, int ox, int oy) : solucio_(n * n, -1), n_(n) {
        <<B>>
        solucio_[0] = pos2num(posicio(ox, oy));
        backtrack(1);
    }

    bool hi_ha_solucio() const {
        <<C>>
    }

    void backtrack(int k) {
         if (<<D>>) { hihasol_ = true; return; }
         int pold = solucio_[k-1];
         for (int d = 1; d <= 8; ++d) {
            <<E>> // int pnew = ...
            if (<<F>>) {
                solucio_[k] = pnew;
                <<G>>
                backtrack(k + 1);
                <<H>>
            }
         }
    }
```

Cognoms, Nom | D.N.I.

(Continueu responent aquí a la Pregunta 4.)