

Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona

Cognoms, Nom

D.N.I.

[illegible]**Titulació:** EI/ETIG

Curs: Q2 2006-2007 (2^{on} Parcial)

Assignatura: Anàlisi i Disseny d'Algorismes

Data: 25 de juny de 2007

Duració: 2 hores

1. **(3 punts)** Es vol emmagatzemar en un dispositiu de capacitat L un conjunt de n arxius. La talla de l'arxiu i -èsim ve donada per $\ell[i]$, i se sap que $\sum_{i=1}^n \ell[i] > L$. Implementeu en C++ o pseudocodi un algorisme voraç que seleccioni un subconjunt dels arxius tot maximitzant el nombre d'arxius que es poden emmagatzemar al dispositiu. Justifiqueu la correctesa del vostre algorisme i analitzeu el seu cost.

SOLUCIÓ:

[illegible]

--	--	--	--	--	--	--	--

(Continueu responent aquí a la Pregunta 1.)

- Utilitza l'esquema que es dona tot seguit, completant els espais marcats i implementant totes les operacions auxiliars que et facin falta.

--

```
}
```

```
bool hi_ha_solucio() { return _hi_ha_solucio; }
```

```
vector<color> solucio() {  
    if (_hi_ha_solucio) return _solucio;  
    else return vector<color>(_N, undef);  
}
```

```
private:
```

```
    int _N;  
    vector<aposta> _apostes;  
    bool _hi_ha_solucio;  
    vector<color> _solucio;  
    void backtracking(int k);  
    // afegiu mètodes i atributs privats aquí, si ho necessiteu
```

```
};
```

```
void concurs::backtrack(int k) {
```

```
    if (  ) {  
        _hi_ha_solucio = true;  
        return;
```

```
    }
```

```
    _solucio[k] = blanca;
```

```
    if (  )
```

```
        backtrack(k + 1);
```

```
    if (  ) {
```

```
        _solucio[k] = negra;
```

```
        if (  )
```

```
            backtrack(k + 1);
```

```
    }
```

```
}
```

SOLUCIÓ:

[illegible]

--	--	--	--	--	--	--	--

(Continueu responent aquí a la Pregunta 2.)

Cognoms, Nom																				D.N.I.				

3. (**3 punts**) S'anomena *arbitratge* a l'ús de les discrepàncies en les taxes de canvi de divises per a obtenir un benefici. Per exemple, durant un breu espai de temps pot succeir que un euro (1 €) valgui 0.95 dòlars, un dòlar (1 \$) valgui 0.75 lliures esterlines i una lliura esterlina (1 £) valgui 1.45€. Llavors, començant amb 1 € podem acabar amb:

$$1\text{€} \times \frac{0.95\$}{1\text{€}} \times \frac{0.75\text{£}}{1\$} \times \frac{1.45\text{€}}{1\text{£}} = 1.033125\text{€},$$

amb un benefici net d'una mica més del 3.3%. Supposeu que hi ha n tipus diferents de divises i que tenim una matriu $C[1..n, 1..n]$ on $C[i, j]$ és la taxa de canvi de (una unitat de) la divisa i a la divisa j . Per exemple si l'euro és la divisa 1 i la lliura esterlina la divisa 2 llavors $C[2, 1] = 1.45$. Es compleix que $C[i, j] = 1/C[j, i]$ per a tota i i j , i que $C[i, i] = 1$.

Implementa en C++ o pseudocodi un algorisme de programació dinàmica que donada la matriu C i l'identificador i d'una divisa, $1 \leq i \leq n$, ens proporcioni el valor del millor arbitratge, on el millor arbitratge és una seqüència $\langle i_1, i_2, \dots, i_k \rangle$ d'identificadors diferents entre sí i diferents d' i tal que

$$C[i, i_1] \cdot C[i_1, i_2] \cdots C[i_{k-1}, i_k] \cdot C[i_k, i]$$

és màxim. Analitza el cost del teu algorisme.

SOLUCIÓ:

[illegible]

--	--	--	--	--	--	--	--

(Continueu responent aquí a la Pregunta 3.)