Solució de l'Examen Final EDA 07/01/2016

Proposta de solució al problema 1

(a) Les respostes són:

	Cert	Fals	Obert
SOR és a la classe P	X		
SOR és a la classe NP	X		
SOR és NP-difícil			X
SAT és a la classe P			X
SAT és a la classe NP	X		
SAT és NP-difícil	X		
Es pot reduir polinòmicament SOR a COL	X		
Es pot reduir polinòmicament COL a SOR			X
No es pot reduir polinòmicament SAT a SOR			X
Es pot reduir polinòmicament COL a SOR, i no es pot reduir polinòmicament SAT a SOR		X	

(b) Si x és l'element $v[i]$, aleshores l'algorisme té cost $\Theta(i)$. Per tant el cost mig és:

$\sum_{0\leq i<n}$ Prob$(x = v[i]) \cdot$ Cost$(x = v[i]) =$

$\sum_{0\leq i<n}$ Prob$(x = v[i]) \cdot \Theta(i) =$

$\Theta(\sum_{0\leq i<n}$ Prob$(x = v[i]) \cdot i) =$

$\Theta(\sum_{0\leq i<n} \frac{i}{n}) =$

$\Theta(\frac{1}{n}) \cdot \Theta(\sum_{0\leq i<n} i) =$

$\Theta(\frac{1}{n}) \cdot \Theta(n^2) =$

$\Theta(n)$

(c) Donat un graf dirigit $G = (V, E)$ amb pesos, l'algorisme de Floyd-Warshall calcula a la vegada el cost del camí mínim de tot vèrtex $u \in V$ a tot vèrtex $v \in V$. En el cas pitjor, el seu cost en temps és $\Theta(|V|^3)$, i en espai $\Theta(|V|^2)$.

Proposta de solució al problema 2

(a) 42 42 23 12 12 12

(b) A cada iteració escriu l'element més petit de $V[0..i]$.

(c) El cost de la primera iteració és constant. Respecte a la resta d'iteracions, el cost està dominat per la inserció en S. Si $i > 0$, a la i-èsima iteració, en què S té inicialment i elements, en el cas pitjor aquesta inserció té cost $\Theta(\log i)$. Si a cada iteració es dóna aquest cost, aleshores el cost total és:

$$\Theta(1) + \sum_{i=1}^{n-1} \Theta(\log i) = \Theta(1) + \Theta(n \log n) = \Theta(n \log n),$$

donat que de la fórmula d'Stirling tenim que $\sum_{i=1}^{n-1} \Theta(\log i) = \Theta(n \log n)$.

(d) En cas que el vector tingui elements repetits, el codi escriu el mateix.

(e) Una possible solució:

```
int min = V[0];
for (int i = 1; i < n; ++i) {
    if (V[i] < min)
      min = V[i];
    cout << min << ' ';
}
```

Cada iteració del bucle només requereix operacions de cost constant. Com que es fan $n-1$ voltes, el cost del codi en funció de n és $\Theta(n)$, que és estrictament millor que $\Theta(n \log n)$.

Proposta de solució al problema 3

Una possible solució al problema:

```
void top_sorts_rec (int k, const Graph& G, vector<int>& sol, vector<int>& indeg) {
  int n = sol.size();
  if (k == n)
    print_solution(sol);
  else
    for (int x = 0; x < n; ++x)
      if (indeg[x] == 0) {
        indeg[x] = −1;
        for (int y : G[x]) −−indeg[y];
        sol[k] = x;
        top_sorts_rec(k+1, G, sol, indeg);
        for (int y: G[x]) ++indeg[y];
        indeg[x] = 0;
      }
}

void top_sorts (const Graph& G, vector<int>& sol) {
  int n = G.size();
  vector<int> indeg(n, 0);
```

```
    for (int x = 0; x < n; ++x)
      for (int y: G[x])
        ++indeg[y];
    top_sorts_rec (0, G, sol , indeg);
}
```

Proposta de solució al problema 4

(a) Omplim la matriu de clausura, diguem-ne *Gstar*, de la manera següent. Des de cada vèrtex u del graf es fa una DFS o una BFS, usant $Gstar[u]$ com a vector de marques booleanes. D'aquesta manera, després de processar el vèrtex u tenim marcats a $Gstar[u]$ aquells vèrtexs als quals es pot arribar des de u. Cal fer doncs n recorreguts, cadascun dels quals costa en el cas pitjor $\Theta(n^2)$. En total el cost en el cas pitjor és $\Theta(n^3)$.

(b) Per inducció. Suposem que $k = 0$. Aleshores $(I + G)^k = I$. El resultat és cert perquè un camí buit només pot connectar un vèrtex amb sí mateix.

Considerem ara el cas que $k > 0$. Tenim que:

$$(I + G)^k_{uv} = \sum_{0 \leq w < n} (I + G)^{k-1}_{uw} \cdot (I + G)_{wv}$$

Suposem que hi ha un camí de u a v amb com a molt k arestes. Si aquest camí té com a molt $k - 1$ arestes, aleshores per hipòtesi d'inducció $(I + G)^{k-1}_{uv} > 0$; com que a més $(I + G)_{vv} > 0$, necessàriament tenim $(I + G)^k_{uv} > 0$. Si en canvi aquest camí té exactament k arestes, aleshores hi ha un vèrtex intermig w tal que hi ha un camí de u a w amb $k - 1$ arestes, i una aresta de w a v. De nou per hipòtesi d'inducció, tenim $(I + G)^{k-1}_{uw} > 0$, i $(I + G)_{wv} > 0$. Per tant, $(I + G)^k_{uv} > 0$ altra vegada.

Suposem ara que no hi ha camí de u a v amb com a molt k arestes. En particular, per tot w tal que $0 \leq w < n$, o bé no hi ha camí de u a w amb com a molt $k - 1$ arestes, o bé no hi ha aresta de w a v; per tant, $(I + G)^{k-1}_{uw} = 0$ o $(I + G)_{wv} = 0$. De forma que $(I + G)^k_{uv} = 0$.

(c) Hi ha un camí del graf de u a v si i només si hi ha un camí del graf de u a v amb com a molt $n - 1$ arestes. Però per l'apartat b), hi ha un camí en el graf de u a v amb com a molt $n - 1$ arestes si i només si $(I + G)^{n-1}_{uv} \neq 0$. Per tant, un algorisme consisteix en calcular $(I + G)^{n-1}$ combinant l'algorisme d'exponenciació ràpida amb l'algorisme d'Strassen, i canviar en la matriu resultant tots els coeficients diferents de 0 per 1. El cost és el de $\Theta(\log n)$ multiplicacions de matrius, cadascuna de les quals costa $\Theta(n^{\log_2 7})$ (ja que les operacions aritmètiques amb enters tenen cost $\Theta(1)$). Així doncs, el cost total és $\Theta(n^{\log_2 7} \log n)$, que és millor que $\Theta(n^3)$.