Cognoms **Nom** **DNI**

Examen Final EDA **Duració: 3h** **07/06/2021**

- *L'enunciat té 4 fulls, 8 cares, i 4 problemes.*
- *Poseu el vostre nom complet i número de DNI a cada problema.*
- *Contesteu tots els problemes en el propi full de l'enunciat a l'espai reservat.*
- *Llevat que es digui el contrari, sempre que parlem de cost ens referim a cost asimptòtic en temps.*
- *Llevat que es digui el contrari,* **cal justificar les respostes.**

Problema 1 **(3 pts.)**

Responeu a les següents preguntes:

(a) (1 pt.) Considereu les funcions següents: $n\log(n^2)$, n^2, $n\log n$, $n(\log n)^2$. Ordeneu-les, de més petita a més gran, segons el seu creixement asimptòtic. Si dues o més funcions tenen el mateix grau de creixement, indiqueu-ho. No cal que raoneu la vostra resposta.

(b) (1 pt.) Raoneu si l'afirmació següent és certa, falsa o no se sap: "Hi ha més problemes NP-complets que pertanyen a P que problemes NP-complets que no hi pertanyen".

(c) (1 pt.) Considereu el codi següent (que no calcula res en particular):

```
int a;
int f (const vector<int>& v, int l, int r) {
  if (l ≥ r) return 0;
  int tot = 0;
  int m = (l+r)/2;
  for (int i = 0; i < a; ++i) {
    if (i%2 == 0) tot += f(v,l,m);
    else          tot += f(v,m+1,r);
  }
  for (int i = l; i ≤ r; ++i)
    for (int j = i+1; j ≤ r; ++j)
      tot += v[i]*v[j];
  return tot;
}
```

Si v és un vector de mida n, determineu de forma raonada el cost d'una crida $f(v, 0, n-1)$ en funció d'n segons el valor de la constant $a \geq 0$, que és un enter que no canvia durant l'execució del programa.

Cognoms Nom DNI

Problema 2 (2.5 pts.)

Un *graf signat* és un graf **no dirigit** i **connex** on cada aresta té una etiqueta $+1$ o -1. S'utilitzen àmpliament en psicologia i ciències socials per a modelitzar les relacions entre persones, grups, corporacions, nacions, etc.

Diem que un graf signat és *estable* si tot cicle del graf té signe positiu. El signe d'un camí o un cicle en el graf és el producte de les etiquetes de les arestes que formen el camí o cicle. Per exemple, un triangle amb tres arestes positives, o amb dues arestes negatives i una positiva és estable, mentre que un triangle amb tres arestes negatives o dues positives i una negativa no ho és.

Un teorema clàssic de la teoria de grafs signats estableix que un graf signat és estable si i només si el conjunt de vèrtexs es pot dividir en dos equips A i B, possiblement buits, de manera que tota aresta positiva connecta vèrtexs del mateix equip, i tota aresta negativa connecta vèrtexs d'equips oposats.

a) (2 pts.) Ompliu les caselles del codi següent per tal de determinar si un graf signat G és o no estable.

```
const int A = 0; const int B = 1; const int no_team = −1;

struct edge {
  int target ; // id of other vertex
  int sign;    // -1 or +1
};

typedef vector<vector<edge>> signed_graph;

int opposed_team(int b) {return (b == A ? B : A);}

int expected_team(int t , int sign) {
  if (sign == 1) return t;
  else return opposed_team(t );
}

bool is_stable (const signed_graph& G, int u, vector<int>& team) {
  for (auto& e : G[u]) {
    int e_t = expected_team(team[u],e.sign);
    if ( [                                        ] ) return false;
    if (team[e. target ] == no_team) {
      [                                        ] ;
      if ( [                              ] ) return false;
    }
  }
  return [            ] ;
}
```

```
bool is_stable (const signed_graph& G, vector<int>& team) {
  team[0] = A;
  return is_stable (G,0,team);
}


int main() {
  int n, m;
  cin >> n >> m;
  signed_graph G(n);
  for (int i = 0; i < m; ++i) {
    int v1, v2, sign;
    cin >> v1 >> v2 >> sign;
    G[v1].push_back({v2,sign});
    G[v2].push_back({v1,sign});
  }
  vector<int> team(n,no_team);
  cout << is_stable (G,team) << endl;
}
```

b) (0.5 pts). Per què hem assignat l'equip A al vèrtex 0 i ni tan sols hem provat l'altra possibilitat?

Cognoms **Nom** **DNI**

Problema 3 **(2.25 pts.)**

En una empresa extraordinàriament jeràrquica i paranoica, tenim n treballadors identificats amb els nombres $\{1, 2, \cdots, n\}$. Aquests nombres també determinen l'ordre jeràrquic dels treballadors. Més concretament, si $i < j$ aleshores i és superior jeràrquicament a j.

Els treballadors només poden transmetre missatges a col·legues que siguin inferiors jeràrquicament. No obstant, s'han d'utilitzar canals de transmissió extraordinàriament segurs i amb un alt cost. L'empresa no vol crear $\Theta(n^2)$ canals de transmissió, sinó que com a molt en vol $O(n \log n)$. Un conjunt de canals de transmissió és *vàlid* si, per a tot parell de treballadors (i, j) amb $i < j$ o bé existeix un canal $i \to j$ o bé existeix un altre treballador k (amb $i < k < j$) que pot actuar com a intermediari. És a dir, existeixen canals $i \to k$ i $k \to j$. Recalquem que no es permeten dos o més salts d'intermediaris. És a dir, $i \to k, k \to l, l \to j$ no és suficient perquè i es pugui comunicar de forma vàlida amb j.

Per exemple, si $n = 8$, una possible solució és:

$1 \to 2$	$2 \to 3$	$7 \to 8$	$5 \to 6$	$6 \to 7$
$6 \to 8$	$1 \to 4$	$2 \to 4$	$3 \to 4$	$4 \to 5$
$4 \to 6$	$4 \to 7$	$4 \to 8$		

Si 5 vol parlar amb 8 haurà d'utilitzar 6 com a intermediari. En canvi, si 4 vol parlar amb 7 ho podrà fer directament.

(a) (1.5 pts.) Dissenyeu un algorisme que determini un conjunt vàlid de canals de transmissió de mida com a molt $O(n \log n)$. No cal que doneu codi, amb una descripció d'alt nivell serà suficient.

(*Pista:* partiu els treballadors en dues meitats. Considereu tres tipus de transmissions: entre treballadors de la primera part, entre treballadors de la segona part i entre treballadors de parts diferents.)

(b) (0.75 pts.) Raoneu per què l'algorisme anterior construeix com a molt $O(n \log n)$ canals de transmissió.

Cognoms **Nom** **DNI**

Problema 4 **(2.25 pts.)**

Volem comprar un cotxe i el fabricant ens ofereix un conjunt d'n equipaments $E = \{1, 2, \cdots, n\}$ que hi podem instal·lar. Hi ha un conjunt d'equipaments $V \subseteq E$ que volem i un conjunt d'equipaments $N \subseteq E$ que no volem de cap de les maneres. Evidentment, $V \cap N = \emptyset$. Malauradament, el fabricant ens imposa dos tipus de restriccions. D'una banda, hi ha un conjunt $I \subseteq E \times E$ de parelles d'equipaments que no es poden instal·lar alhora. D'altra banda, hi ha un conjunt de relacions entre equipaments $R \subseteq E \times E$. Una parella $(i, j) \in R$ significa que si volem instal·lar i també cal instal·lar j. Necessitem decidir si és possible satisfer les restriccions imposades pels conjunts V, N, I i R. Per exemple, si $n = 8$ i les restriccions són:

- $V = \{1, 4\}$
- $N = \{2, 3\}$
- $I = \{ (1, 3), (2, 3), (1, 5), (5,8)\}$
- $R = \{ (1, 6), (4, 1), (6, 7)\}$

aleshores les restriccions es poden satisfer tot instal·lant, per exemple, els equipaments $\{1, 4, 6, 7, 8\}$. Notem que no és el conjunt mínim d'equipaments que satisfà totes les restriccions.

(a) (1.5 pts.) Construeix una reducció polinòmica del problema que acabem d'introduir cap a $2SAT$ (determinar si una CNF en lògica proposicional on cada clàusula té **com a molt** 2 literals és satisfactible). No cal demostrar que la reducció compleix les propietats desitjades.

Pista: utilitza variables Booleanes p_i, amb $1 \leq i \leq n$, que signifiquen que l'equipament i s'instal·larà.

(b) (0.75 pts.) Podem afirmar que el problema dels equipaments es pot resoldre en temps polinòmic? Només valorarem de forma positiva argumentacions concises i precises.