Solució de l'Examen Final EDA 07/06/2021

Proposta de solució al problema 1

(a) Notem que $n\log(n^2) = 2n\log n$, i per tant, aquest funció té el mateix grau de creixement que $n\log n$. De fet, aquestes dues són les que tenen menor grau de creixement, seguides per $n(\log n)^2$ i posteriorment per n^2.

Tot i que no se'ns demanava la justificació, l'adjuntem:

$$\lim_{x\to\infty} \frac{n\log n}{n(\log n)^2} = \lim_{x\to\infty} \frac{1}{\log n} = 0$$

Per tant, $n\log n$ és menor asimptòticament parlant.

De forma similar:

$$\lim_{x\to\infty} \frac{n(\log n)^2}{n^2} = \lim_{x\to\infty} \frac{(\log n)^2}{n} = 0$$

(b) Sabem que, o bé tots els problemes NP-complets pertanyen a P o cap d'ells hi pertany. No obstant, no se sap quina de les dues situacions és la correcta. Per tant, no sabem si l'afirmació que ens fan és certa o falsa.

Si fos certa, necessàriament tots els problemes NP-complets pertanyerien a P. Si fos falsa, cap d'ells ho faria.

(c) Ens fixem que quan no estem al cas base el primer bucle s'executa a vegades, on a cada volta es fa una crida recursiva de mida la meitat. En acabar el primer bucle, passem a fer un treball (els dos bucles ennierats) que té cost $\sum_{i=1}^{n-1}\Theta(i)$, que equival a $\Theta(n^2/2)$ i per tant $\Theta(n^2)$. Així doncs, el cost d'aquesta funció recursiva es pot descriure per:

$$T(n) = a \cdot T(n/2) + \Theta(n^2)$$

d'on identifiquem $b = 2$, $k = 2$ i $\alpha = \log_2 a$. Aplicant el teorema mestre de recurrències divisores podem afirmar que

- Si $a < 4$, aleshores $\alpha < k$ i per tant, $T(n) \in \Theta(n^2)$
- Si $a = 4$, aleshores $\alpha = k$ i per tant, $T(n) \in \Theta(n^2 \log n)$
- Si $a > 4$, aleshores $\alpha > k$, i per tant, $T(n) \in \Theta(n^{\log_2 a})$

Proposta de solució al problema 2

(a)
```
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 (team[e.target] ≠ noTeam and team[e.target] ≠ e_t) return false;
    if (team[e.target] == noTeam) {
      team[e.target] = e_t;
      if (not is_stable (G, e.target, team)) return false;
    }
  }
  return true;
}
```

(b) Ho podem fer perquè si existeix una manera correcta de dividir els vèrtexs en equips A i B, aleshores segur que existeix una manera correcta on el vèrtex 0 va a l'equip A. Només cal adonar-se que, donada una solució on 0 va a l'equip B, si canviem l'equip de tots els vèrtexs, obtindrem una solució correcta on 0 va a l'equip A.

Proposta de solució al problema 3

(a) Dissenyarem una funció recursiva `canals(int l, int r)` que escriurà un conjunt vàlid de canals de transmissió per als treballadors $\{l, l+1, \cdots, r-1, r\}$. Si $l = r$ només hi ha un treballador i per tant no cal cap canal.

En cas contrari, prenem m el punt mig entre l i r, i considerem $n = r - l + 1$. Això ens parteix els treballadors en dues meitats d'aproximadament $n/2$ elements. Primer escriurem, amb una crida recursiva, tots els canals necessaris per comunicar els treballadors $\{l, l+1, \cdots, m-1\}$ i a continuació, amb una altra crida recursiva, els canals necessaris per comunicar els treballadors $\{m+1, m+2, \cdots, r\}$. Fixem-nos que les úniques comunicacions que ens falten són d'entre treballadors de meitats diferents (i també d'entre treballadors de l'esquerra cap a m). Per tal de comunicar les dues meitats, crearem els canals següents:

- $l \to m,\ l+1 \to m,\ \ldots,\ m-1 \to m$
- $m \to m+1,\ m \to m+2,\ \ldots,\ m \to r-1,\ m \to r$

Intuïtivament, m és l'intermediari per les comunicacions entre treballadors de l'esquerra i treballadors de la dreta. Notem, a més, que hem creat exactament $n-1$ canals addicionals.

Adjuntem codi per més concreció, tot i que no es demanava.

```
void canals (int l, int r) {
  if (l >= r) return;
  else {
    int m = (l+r)/2;
    canals(l,m-1);
    canals(m+1,r);
    for (int k = l; k < m; ++k)
      cout << k << " --> " << m << endl;
    for (int k = m+1; k <= r; ++k)
      cout << m << " --> " << k << endl;
  }
}

int main() {
  int n; cin >> n;
  canals (1,n);
}
```

(b) Fixem-nos que el nombre de canals que crea la nostra funció es pot descriure per la recurrència:

$$C(n) = 2C(n/2) + \Theta(n)$$

que, pel teorema mestre de recurrències divisores té solució $C(n) \in \Theta(n \log n)$.

Proposta de solució al problema 4

(a) Construirem una instància de $2SAT$ que serà la conjunció de les següents clàusules:

- Per a cada $i \in V$, una clàusula p_i.
- Per a cada $i \in N$, una clàusula $\neg p_i$.
- Per a cada $(i, j) \in I$, una clàusula $\neg p_i \vee \neg p_j$.
- Per a cada $(i, j) \in R$, una clàusula $\neg p_i \vee p_j$.

(b) Com que hem construït una reducció polinòmica del problema dels equipaments cap a $2SAT$, i sabem que $2SAT$ és un problema que es pot resoldre en temps polinòmic, aleshores podem concloure que el problema dels equipaments també es pot resoldre en temps polinòmic.

Tot i que no es demanava, remarquem que l'algorisme polinòmic pot consistir en calcular primer la reducció de l'apartat anterior i aplicar un algorisme polinòmic per $2SAT$ sobre la fórmula resultant.