
Algorísmia i Programació III

– PROJECTE 2019 – Fantasy Football

Des de principis dels 90, diverses entitats relacionades amb el món de l'esport han estat organitzant competicions de Fantasy Football, cadascuna amb les seves petites diferències. Grosso modo, el joc té com a referència un competició futbolística real. Els participants del joc han de confeccionar un equip amb jugadors reals, que aniran rebent més o menys punts al llarg de la competició d'acord amb el seu rendiment en aquesta. Al final de la temporada, guanyarà el participant que hagi obtingut el major nombre de punts.

En aquest projecte se us demana implementar un programa de suport per a jugar a Fantasy Football. A tal fi disposem de la llista de tots els jugadors de la Lliga de Futbol Professional d'Espanya. Per cada jugador coneixem el seu nom, la posició, el preu, l'equip al qual pertany i la puntuació que va obtenir l'any anterior. L'objectiu del vostre programa és formar l'equip que maximitzi el nombre de punts que va aconseguir l'any anterior. No obstant, no podem escollir tots els jugadors que vulguem, ja que tenim un límit total de T euros i també un límit de J euros per cada jugador que escollim. Finalment, la tàctica a desplegar ens ve donada per tres enters N_1, N_2 i N_3 , que indiquen que hem d'escollir N_1 defenses, N_2 migcampistes i N_3 davanter, on $1 \leq N_1, N_2, N_3 \leq 5$ i $N_1 + N_2 + N_3 = 10$. Com és d'esperar, a més a més, sempre hem d'alinear un porter.

Les dades d'entrada per al vostre programa s'organitzen en dos fitxers, un amb la *base de dades de jugadors*, i un altre *arxiu de consulta* amb les restriccions de pressupost i la tàctica desitjada:

Format de la base de dades de jugadors:

Cada línia de l'arxiu representa la informació associada a un jugador. Vindrà amb la forma:

Nom;Posició;Preu;Equip;Punts

- Nom: un *string* que pot contenir espais ens blanc, però mai el caràcter ';'.
- Posició: tres caràcters d'entre les següents 4 possibilitats: "por", "def", "mig", "dav".
- Preu: un nombre natural
- Equip: un *string* que pot contenir espais ens blanc, però mai el caràcter ';'.
- Punts: un nombre natural

Garantim que sempre hi haurà prou jugadors per confeccionar l'alineació i que per cada posició hi ha almenys 5 jugadors que tenen un preu de 0 euros.

Format de l'arxiu de consulta:

Contindrà 5 naturals en aquest ordre: N_1 N_2 N_3 T J , amb el significat explicat en la introducció del projecte.

Un exemple d'aquests dos arxius el teniu a continuació:

Base de dades de jugadors:

```
Ter Stegen;por;41843856;FC Barcelona;239
Arias;def;3752802;Atlético De Madrid;130
Sergio León;dav;3278997;Levante UD;35
De Marcos;def;2883946;Athletic Club;112
Granero;mig;963151;RCD Espanyol;130
```

Consulta:

```
3 4 3
75000000
20000000
```

Format de sortida:

L'arxiu contindrà 7 línies:

1. Un double (amb com a molt 1 decimal) que indicarà els segons que han estat necessaris per trobar aquesta solució.
2. Els cinc caràcters "POR: ", seguits del nom del porter escollit.
3. Els cinc caràcters "DEF: ", seguits dels noms dels defenses escollits, separats amb ','
4. Els cinc caràcters "MIG: ", seguits dels noms dels migcampistes escollits, separats amb ','
5. Els cinc caràcters "DAV: ", seguits dels noms dels davanterers escollits, separats amb ','
6. Els set caràcters "Punts: ", seguits dels punts de l'equip
7. Els sis caràcters "Preu: ", seguits del preu de l'equip

Un exemple concret de sortida el podeu veure a continuació:

```
7.2
POR: David Soria
DEF: Djené;Hermoso;Siovas;Raúl García;Rubén Peña
MIG: Campaña;Maksimovic;Granero;Pere Pons
DAV: Jorge Molina
Punts: 2016
Preu: 73376753
```

Tasques a realitzar:

El projecte s'ha de realitzar en parelles. Heu d'entregar tres arxius (i només tres!) en C++, amb **exactament** el noms *exh.cc*, *greedy.cc* i *mh.cc* a través del Racó. No acceptarem arxius comprimits. Els tres programes rebran tres arguments per línia de comandes: el primer d'ells és el nom de l'arxiu que conté la base de dades de jugadors, el segon és el nom de l'arxiu de consulta i el tercer és el nom del fitxer de sortida.

- *exh.cc*: ha d'implementar una cerca exhaustiva. Cada vegada que obtingui una solució millor, l'ha de sobreesciure a l'arxiu de sortida. És a dir, volem que si avortem el programa, dins l'arxiu de sortida hi hagi la millor solució trobada fins al moment. La data límit d'entrega és el 15 de novembre.
- *greedy.cc*: ha d'implementar un algorisme golafre. Degut a la dificultat computacional del problema, no s'espera que obtingui una solució òptima. El que sí demanarem és que sigui ràpid: ha de ser instantani fins i tot en les instàncies més grans que proporcionem. La data límit d'entrega és el 13 de desembre.
- *mh.cc*: ha d'implementar una metaheurística de les explicades a classe (excepte Basic Local Search). Com amb la cerca exhaustiva, volem que cada vegada que obtingui una solució millor, la sobreescrigui a l'arxiu de sortida. La data límit d'entrega és el 7 de gener.

Criteris d'avaluació:

Dels 2 punts del projecte, la meitat tindran en compte la llegibilitat del codi i l'altra meitat la qualitat de les solucions obtingudes i el temps necessari per a calcular-les. Us pot ser útil saber que no deixarem executar la cerca exhaustiva més de 5 minuts, ni l'algorisme greedy més de 5 segons, ni la metaheurística més d'1 minut. Avaluarem el vostre codi sobre un joc de proves privat on els exemples són de mida similar al joc de proves públic que us facilitem. Per tal de comprovar que les vostres solucions són correctes podeu utilitzar el checker que us facilitem. Recordem que el checker en cap cas comprova que la solució sigui òptima: tan sols comprova que el format de l'arxiu sigui correcte i que els punts i el preu que hi apareixen estan ben calculats.