Poseu el nom a tots els fulls
Entregueu els problemes en fulls separats
Les respostes han de ser justificades

Problema 1 *7 punts* Temps estimat: 60 min

Una botiga vol conèixer quina és l'hora d'afluència màxima de clients al llarg d'un dia. Per a fer-ho col·loca dos sensors, un a cada porta del local. Cada sensor compta quanta gent entra i surt cada minut. Cada sensor guarda els resultats en un fitxer, `sensor1.txt` i `sensor2.txt` respectivament. Les dades que guarden els sensors tenen següent format:

$$H_1\ M_1\ E_1\ S_1\ H_2\ M_2\ E_2\ S_2\ \ \ldots\ \ -1\ -1\ E_s\ S_s$$

on l'enter H_i és l'hora, l'enter M_i el minut, l'enter E_i és el nombre de persones que han entrat en el moment $H_i\ M_i$ i l'enter S_i és el nombre de les que han sortit en aquest minut. Les dades s'enregistren ordenades temporalment en els fitxers. Els sensors només enregistren una dada al seu fitxer quan detecta una entrada o sortida de clients (és a dir, no escriu un minut si no ha entrat ni ha sortit ningú). El primer registre $H_1\ M_1\ E_1\ S_1$ del fitxer correspon a la primera hora del dia que entren clients a la botiga. El darrer registre és una marca de final que es caracteritza perquè tant l'hora com el minut valen -1.

Es disposa de:

La definició del tipus *Registre* que representa cada un dels registres que hi ha en els fitxers donats, es a dir, representa la informació de cada minut i de la definició del tipus *Hora* que representa un moment (hora i minuts). A més, ens donen operacions que treballen amb objectes dels tipus definits.

tipus
Registre
Hora
ftipus

acció *LlegirRegistreFST*(**entsor** $f : FST$, **sor** r : *Registre*)
{**Pre:** f es un FST de registres (H_i, M_i, E_i, S_i), i $mode(f) = lectura \ \wedge\ pe(f) = \alpha\ \wedge\ en\_curs(f) = \beta\ \wedge\ pd(f) = \gamma \cdot \delta\ \wedge\ pd(f) \neq \emptyset$}
{**Post:** $mode(f) = lectura \ \wedge\ pe(f) = \alpha.\beta\ \wedge\ en\_curs(f) = \gamma\ \wedge\ pd(f) = \delta\ \wedge$ r representa el registre $\gamma\ \ (H_i, M_i, E_i, S_i)$}

funció *RegistreSentinella*(**ent** r : *Registre*) **retorna booleà**
{**Pre:** }
{**Post:** RegistreSentinella(r) és cert si tant l'hora com el minut representat en r són -1. És fals en cas contrari}

funció *MovClients*(**ent** r : *Registre*) **retorna enter**
{**Pre:** r representa (H, M, E, S)}
{**Post:** $MovClients(r) = E - S$}

funció *HoraRegistre*(**ent** r : *Registre*) **retorna** *Hora*
{**Pre:** r representa (H, M, E, S)}
{**Post:** $Hora(r)$ representa (H, M)}

acció *EscriureHora*(**ent** h : *Hora*)
{**Pre:** h representa $(H, M) \wedge CSS = \gamma$}
{**Post:** $CSS = \gamma\{H, M\}$}

funció *EsAbans*(**ent** $h1, h2$: *Hora*) **retorna booleà**
{**Pre:** }
{**Post:** $EsAbans(h1, h2)$ és cert si l'hora representada per $h1$ és anterior a la representada per $h2$. En cas contrari és fals.}

funció *Alhora*(**ent** $h1, h2$: *Hora*) **retorna booleà**
{**Pre:** }
{**Post:** Alhora(h1, h2) és cert si h1 i h2 representen la mateixa hora. En cas contrari és fals.}

Es demana:

- *(5 punts)* Dissenyeu un algorisme que a partir de les dades dels dos FST escrigui pel CES l'hora (hora i minut) de màxima afluència, és a dir, el moment en que el nombre de clients que hi ha a la botiga és màxim, així com el nombre de clients que hi ha hagut en aquest moment. Suposeu que cap de les dues seqüències dels dos fitxers és buïda.
- *(1 punt)* Definir el tipus Registre de manera raonada.
- *(1 punt)* Definir el tipus Hora i dissenyeu els subprogrames $EscriureHora(\textbf{ent}\ h1 : Hora)$ i $EsAbans(\textbf{ent}\ h1, h2 : Hora)$

Problema 2 — *3 punts* — Temps estimat: 30 min

Ens proporcionen el tipus *tauEnters* i el següent algorisme que l'utilitza:

```
tipus
  tauEnters = tupla
                l : enter;  {La taula està ocupada entre les posicions 1 i l. (1 ≤ l ≤ 100000)}
                t : taulaEnters ftupla
  taulaEnters = taula [1..100000] de enter
ftipus

1  acció ordena(entsor te : tauEnters)
2    var
3      i, j : enter
4    fvar
5    {Pre: te = TE}
6    i := te.l;
7    mentre i > 1 fer
8      j := 2
9      mentre j ≤ i fer
10       si te.t[j-1] > te.t[j] →
11                                  aux := te.t[j-1]
12                                  te.t[j-1] := te.t[j]
13                                  te.t[j] := aux
14       □ te.t[j-1] ≤ te.t[j] →
15       fsi
16       j := j+1
17     fmentre
18     i := i-1
19   fmentre
20   {Post: te és una permutació dels elements de la taula t de TE, tal que es cumpleix te.t[i] ≤ t[j] per
     tot 1 ≤ i < j ≤ TE.l i te.l = TE.l}
21 facció
```

Identifiqueu la mida de les dades i calculeu la complexitat asimptòtica d'aquest algorisme.

Ha de constar en el full de resposta tot el desenvolupament que us porta als resultats.