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: 40 min

Observem que els dos fitxers donats, el `"queFem"` i el fitxer `"preferencies"`, estan ordenats segons les dates, per tant, malgrat els registres que contenen són diferents en cada cas, el criteri d’ordenació és el mateix. El fitxer que s’ha de construir contindrà només dates que hi ha al fitxer `"preferencies"`, però per construir `"elsMeusPlans"` s’hauran d’utilitzar els dos fitxers. Per tant, aplicarem l’esquema de Fusió als dos fitxers donats, però acabarem un cop s’hagi arribat al final d’algun dels dos fitxers(és a dir, hi haurà una única iteració).

Com que ens demanen una acció que tingui com a paràmetres d’entrada els noms dels tres fitxers involucrats, farem una acció amb tres paràmetres d’entrada de tipus *Cadena*.

acció $agenda($**ent** $programaActes : Cadena,$ **ent** $agenda : Cadena,$ **ent** $novaAgenda : Cadena)$
  var
    $prog, ag, nAg : FST$
    $dataAg, dataActe : tData$
    $tipusActe, tipusActeAg : tParaula$
    $Acte : tEspectacle$
  fvar
  $nAg := ObrirFST(escriptura, novaAgenda)$
  $ag := ObrirFST(lectura, agenda)$
  $prog := ObrirFST(lectura, programaActes)$
  $llegirActeFST(prog, dataActe, tipusActe, Acte)$
  $llegirDiaAgendaFST(ag, dataAg, tipusActeAg)$
  mentre $\neg(dataSentinella(dataActe) \vee dataSentinella(dataAg))$ **fer**
    si $anteriorData(dataActe, dataAg) \rightarrow$
        $llegirActeFST(prog, dataActe, tipusActe, Acte)$
    $\Box$ $igualData(dataActe, dataAg) \rightarrow$
        si $igualParaules(tipusActe, tipusActeAg) \rightarrow$
            $escriureDataFST(nAg, dataAg)$
            $escriureEspectacleFST(nAg, Acte)$
            $llegirDiaAgendaFST(ag, dataAg, tipusActeAg)$
        $\Box$ $\neg igualParaules(tipusActe, tipusActeAg) \rightarrow$
        fsi
        $llegirActeFST(prog, dataActe, tipusActe, Acte)$
    $\Box$ $anteriorData(dataAg, dataActe) \rightarrow$
        $llegirDiaAgendaFST(ag, dataAg, tipusActeAg)$
    fsi
  fmentre
  $TancarFST(prog)$
  $TancarFST(ag)$
  $escriureDataSentinellaFST(nAg)$
  $TancarFST(nAg)$
facció

Observem que quan es criden els subprogrames d’$ObrirFST$, el que passem són variables de tipus *Cadena*, per tant, la seva sintaxi **no** necessita comes.

Problema 2 — *3 punts* — Temps estimat: 20 min

La mida de les dades és el nombre màxim de xifres de la taula de l’*enterGran x*, $MAXXIFRES$, que anomenarem n.

Sigui $t_i(n)$ la funció de cost d’executar la línia i-éssima, i $t_{i-j}(n)$ la funció de cost d’executar l’acció entre les línies i i j.

$t_{canviaSigne}(n) = t_{16}(n) + t_{18}(n) + max(t_{17}(n), t_{19}(n))$

per la composició alternativa. En cost asimptòtic, aplicant reflexivitat i la regla de la suma:

$t_{canviaSigne}(n) \in O(max(t_{16}(n), t_{18}(n), t_{17}(n), t_{19}(n)))$

$t_{17}(n)$ és el cost d’una acció buida que pertany a $O(1)$. $t_{19}(n)$ és el cost de fer d’una assignació, més l’aplicació d’una negació, i l’accés a una tupla, i per tant, $t_{19}(n) \in O(1)$. Queda

$t_{canviaSigne}(n) \in O(max(t_{16}(n), t_{18}(n)))$

$t_{16}(n) = t_{nXifres(e)}(n) + t_{=}(n) + t_1(n) + t_{\wedge}(n) + t_{.}(n) + t_{[]}(n) + t_{=}(n) + t_0(n)$

Per la regla de la suma i tenint en compte que $t_{=}(n) \in O(1)$, $t_1(n) \in O(1)$, $t_{\wedge}(n) \in O(1)$, $t_{.}(n) \in O(1)$, $t_{[]}(n) \in O(1)$ i, $t_0(n) \in O(1)$,

$t_{16}(n) \in O(max(t_{nXifres(e)}(n), 1, 1, 1, 1, 1, 1, 1))$

$t_{16}(n) \in O(t_{nXifres(e)}(n))$

L’expressió a avaluar en la línia 18 és semblant a la de la línia 16. La única diferència és que cal aplicar l’operador negació a l’expressió avaluada en la línia 16. Com que la negació és de cost constant, tenim també per la regla de la suma que

$t_{18}(n) \in O(t_{nXifres(e)}(n))$

$t_{nXifres(e)}(n) = t_{nXifres} + t_e(n)$

Com que $t_e(n) \in O(1)$, per reflexivitat i la regla de la suma tenim que

$t_{canviaSigne}(n) \in O(t_{nXifres}(n))$

Analitzem la funció $nXifres$:

$t_{nXifres}(n) = t_3(n) + t_4(n) + t_{5-12}(n) + t_{13}(n)$

$t_3(n) \in O(1)$ ja que es tracta d’una assignació i una expressió que accedeix a un valor.

$t_4(n) \in O(1)$ ja que es tracta d’una assignació i una expressió amb una resta.

$t_{13}(n) \in O(1)$ ja que és un retorna i l’expressió és una suma d’un variable i una constant.

Per reflexivitat i la regla de la suma tenim que

$t_{nXifres}(n) \in O(max(1, 1, t_{5-12}(n), 1))$

$t_{nXifres}(n) \in O(t_{5-12}(n))$

Hi ha dues regles per calcular el cost d’un mentre que depenen de si el cos del mentre té el mateix cost a cada iteració o el cost és diferent. Comencem doncs a analitzar el cost del cos del mentre $t_{6-11}(n)$:

$t_{6-11}(n) = t_6(n) + t_{7-11}(n)$

$t_6(n) \in O(1)$ ja que es tracta d’una assignació, un accés al camp d’una tupla, un accés a la taula, i l’aplicació de l’operador relacional $\neq$.

$t_{7-11}(n) \in O(1)$, ja que totes les expressions a avaluar són constants, i la branca pitjor és una assignació amb temps constant (expressió amb resta).

Per reflectivitat i la regla de les sumes, $t_{6-11}(n) \in O(1)$. Amb aquest resultat (el cost del cos és el mateix a cada iteració), podem aplicar la regla del mentre:

$t_{5-12}(n) = f(n) \cdot (t_5(n) + t_{6-11}(n)) + t_5(n))$ on $f(n)$ és el nombre de vegades que s’executa el **mentre**

I com que $_5(n) \in O(1)$ i $t_{6-11}(n) \in O(1)$, tenim que $t_5(n) + t_{6-11}(n)) + t_5(n) \in O(1)$ per la regla de la suma, i per la regla de productes:

$t_{5-12}(n) \in O(f(n)) \cdot O(1)$,

$t_{5-12}(n) \in O(f(n))$

En el cas pitjor el mentre recorre la següent seqüència de valors de i: $MAXXIFRES - 1$, $MAXXIFRES - 2, \ldots, 1$. Així doncs, recorre $MAXXIFRES - 1$, elements. Com que $n = MAXXIFRES$, $f(n) = n - 1$

Per invariança additiva $f(n) \in O(n)$, i per transitivitat ($t_{nXifres}(n) \in O(t_{5-12}(n))$ i $t_{5-12}(n) \in O(n)$):

$t_{nXifres}(n) \in O(n)$

Com que $t_{canviaSigne}(n) \in O(t_{nXifres}(n))$ i $t_{nXifres}(n) \in O(n)$, per transitivitat

$\boxed{t_{canviaSigne}(n) \in O(n)}$