Limes transire

Portem un carregament de caixes, i en travessar la frontera pel pont d'Ilerda, ens demanen que s'han de poder veure les etiquetes de totes les caixes. Les caixes són de diferents alçades i tenen l'etiqueta a la part de dalt. No podem canviar l'ordre de les caixes i hem de treure les caixes que ens molesten i quedar-nos amb les que ens van bé, intentant que quedin el màxim nombre de caixes.

Exemple

Per exemple, portem 5 caixes, amb alçades:

- Caixa 1 amb alçada 2
- Caixa 2 amb alçada 1
- Caixa 3 amb alçada 3
- Caixa 4 amb alçada 2
- Caixa 5 amb alçada 6.

En la solució, la primera ha de tenir l'alçada més baixa i la darrera la més alta. En aquest cas, les solucions amb més nombre de caixes són:

- Les caixes 1, 3 i 5 amb alçades 2, 3 i 6.
- Les caixes 2, 3 i 5 amb alçades 1, 3, i 6.
- Lex caixes 2, 4 i 5 amb alçades 1, 2 i 6.

Totes les solucions tenen llargada 3.

Requeriments

Es demana realitzar:

- Un programa que llegeixi les dades d'un fitxer de text i retorni una seqüència d'alçades creixent i de largada màxima.
- Un informe que contingui, sobre els algorismes principals: pseudo-codi en disseny recursiu i iteratiu, especificació formal, cost teòric i experimental.

Arguments i parametres

L'execució de l'aplicació haurà de seguir la següent sintaxi:

```
$./ilerda [fitxer-caixes]
```

Els arguments opcionals del programa són els següents:

fitxer-caixes : Fitxer d'entrada amb la llista de caixes.

Si no hi ha arguments opcionals, l'aplicació llegeix de l'entrada estàndard i escriu a la sortida estàndard.

El format del fitxer tindrà n tuples separades per `\n` amb el format:

```
contingut-caixa alçada-caixa
```

Avaluació

En l'informe de la pràctica, els algorismes principals, les taules i les gràfiques han d'estar comentats.

Els programes han de compilar sense errors ni warnings i passar els tests amb `make test`, altrament no s'avaluarà.

La baremació de la pràctica (sobre 10) serà la següent:

Especificació 1 punt, especificació formal;

Costos 3 punts, anàlisi de costos teòrics (recursiu i iteratiu) i empírics dels algorismes;

Disseny 3 punts, recursiu i transformació a iteratiu (es tindrà en compte el cost de l'algorisme);

Implementació 3 punts

- 1 punt, recursiu en Haskell
- 1 punt, bones pràctiques de programació
- 1 punt, utilització dels recursos del llenguatge de programació.

Enviament

L'assignació és per parelles, i representa un 20% de la nota final. Presenteu la pràctica al Campus Virtual de la UdL amb dos fitxers: un pdf per a l'informe, i un arxiu comprimit, `tgz` o `zip` (no s'admeten altres formats de compressió), per al codi font.

Els programes que s'hagin de compilar ho han de fer amb:

```
$ make
```

Els programes es testejen fent:

```
$ make test
```

La pràctica podria ser verificada individualment el primer dia de laboratori després de l'entrega.