

Pràctica 1: El problema de l'illa

Implementacions i Millores

illa.py (Python)

La implementació en Python ha estat millorada amb les següents modificacions:

1. **Reorganització del codi:**
 - Implementació orientada a objectes amb la classe `IllaProblemSolver`
 - Millor encapsulament de dades i mètodes
 - Seguiment de les bones pràctiques de pylint
2. **Millores funcionals:**
 - Suport per entrada per consola
 - Gestió d'errors més robusta
 - Decorador per mesurar temps d'execució
 - Millor gestió de fitxers amb encoding UTF-8
3. **Optimitzacions:**
 - Millora en la detecció de cicles
 - Estructura de dades més eficients
 - Reducció de la recursió

illa.hs (Haskell)

La versió Haskell ha estat modificada per incloure:

1. **Millores funcionals:**
 - Implementació de `ListF` com a functor
 - Millor gestió d'entrada/sortida
 - Suport per entrada per consola
2. **Estructures de dades:**
 - Ús de `Map` i `Set` per millor eficiència
 - Implementació de tipus algebraics
 - Millor gestió de la memòria

Anàlisi de rendiment

Cost computacional actualitzat

- **illa.py:**
 - Cas mitjà: $O(n^2)$
 - Pitjor cas: $O(r * n^2)$
 - Millora en l'ús de memòria: $O(n)$
- **illa.hs:**
 - Cas mitjà: $O(n \log n)$
 - Pitjor cas: $O(r * n \log n)$
 - Gestió de memòria lazy: $O(1)$ space leaked

Observacions de rendiment

1. **Millors en Python:**
 - Reducció del temps d'execució en casos mitjans
 - Millor gestió de la memòria
 - Detecció més eficient de cicles
2. **Millors en Haskell:**
 - Implementació més funcional
 - Millor tipus de seguretat
 - Avaluació lazy més eficient

Limitacions conegudes

`illa.py`

- Optimitzat per a casos mitjans
- Cost quadràtic en pitjors casos
- Dependent de l'estructura dels cicles

`illa.hs`

- Overhead del garbage collector
- Cost logarítmic en operacions de Map/Set
- Possibles stack overflows en recursió profunda

Conclusions

1. Ambdues implementacions són més robustes i mantenibles
2. Millor gestió d'errors i casos extrems
3. Codi més net i documentat
4. Millor rendiment en casos típics