Irene Bertolín Rico

Carla Cánovas Iglesias

Adrian Cristian Crisan

Alex Moreno Baeza

**ESTRUCTURES DE DADES**

# Document

HashMap<String, Sheet>

El Key ve representat per el nom del full i el valor és el full associat a aquell nom.

|  | Cost mitjà | Cost pitjor |
| --- | --- | --- |
| Afegir full | Θ(1) | 𝑂(𝑛) |
| Eliminar full | Θ(1) | 𝑂(𝑛) |
| Obtenir full | Θ(1) | 𝑂(𝑛) |
| Obtenir mida | 𝑂(1) | |

# **Nota**: Sent n el número de fulls

# Sheet

HashMap<String, Cell>

El key és la posició de la cel·la dins el full corresponent, i el valor correspon a la cel·la que correspon a la posició del key.

|  | Cost |
| --- | --- |
| Cost de crear un full | Θ(1) |
| Cost d’afegir una fila | 𝑂(𝑛) |
| Cost d’afegir una columna | 𝑂(𝑛) |
| Cost d’eliminar una fila | 𝑂(𝑛) |
| Cost d’eliminar una columna | 𝑂(𝑛) |
| Cost de modificar una cel·la | Θ(1) |

# **Nota**: Sent n el número de cel·les

# Cell

HashSet<Cell> myReferences

Per emmagatzemar les cel·les que referencia la cel·la actual.

HashSet<Cell> referencesToMe

Per emmagatzemar les cel·les que tenen referenciada la cel·la actual.

|  | Cost mitjà | Cost pitjor |
| --- | --- | --- |
| Accedir a una referència | Θ(1) | 𝑂(𝑛) |
| Eliminar una referència | Θ(1) | 𝑂(𝑛) |
| Afegir una referència | Θ(1) | 𝑂(𝑛) |

# **Nota**: Sent n el número de cel·les

**ALGORISMES DE LES FUNCIONALITATS PRINCIPALS**

**Obtenir un valor o contingut**: es dona una posició, d’obté retornant el valor obtingut al fer un get sobre el hashMap, en cas de que contingui la clau (la posició) indicada, en cas contrari es retorna 0, donat que la cel·la té valor per defecte.

**Modificar cel·l**a: la cel·la rep un contingut, aquest pot ser un valor normal de tipus numèric o string o un contingut com una referència d’una altra cel·la o una funció. El parser resol les referències i funcions, afegint les referències per poder saber en el cas de tornar-se a modificar saber quines són les cel·les que s’han d’actualitzar. En el cas d’actualitzar altres cel·les, només s’actualitzen les cel·les que tenen referenciada la cel·la modificada.

El parser:

* Si es una referència, obté el valor fent una crida a la cel·la, que crida a full i si és necessari, perquè la cel·la és d’un altre full, llavors a document i obté el valor.
* Si es una funció, calcula la funció fent una crida a la classe funcions amb els paràmetres necessaris, en el cas de tenir referències es procedeix de la mateixa manera que en el cas anterior, per obtenir els valors de cada cel·la.

Cela: Una vegada obtingut el valor i establert el contingut de la cel·la, es procedeix a actualitzar les referències d’altres cel·les a aquesta, per això iterem sobre el HashSet cridant un mètode anomenat update, que recalcula el contingut de la cel·la utilitzant el parser.

**Afegir files/columnes**: al afegir una fila o columna, es desplacen les cel·les que tenen una posició superior a la de la fila o columna afegida, a més s’actualitzen totes les cel·les que contenen referències. Per comprovar quines cel·les hem de desplaçar agafem la part que correspon a la fila o a la columna i la canviem per la nova, si fa falta. Per actualitzar les referències, es comproven totes les cel·les, i es canvien les que contenen les referències, en el cas de coincidir amb el desplaçament de files o columnes, l'actualització es realitza modificant el contingut de la cel·la.

**Eliminar files/columnes**: Es realitza de la mateixa forma que al afegir files/columnes però en actualitzar les referències si aquestes coincideixen amb les files o columnes eliminades el contingut esdevé 0 i en el cas de no estar referenciada per cap cel·la s’esborra del HashMap del full, donat que la cel·la esdevé una cel·la que no aporta cap informació.

**Copiar cel·la**: s’obté el valor de la cel·la que es desitja copiar i es modifica el valor de l’altre la cel·la.

**Copiar bloc cel·les**: es fa el mateix que al copiar una cel·la però en aquest cas, per tot el bloc de cel·les.

**Moure cel·la**: es clona la cel·la donada una posició, canviant la posició. Per clonar la cel·la implementem la interficie Cloneable, i després, esborrem la cel·la clonada i actualitzem les referències a aquesta cel·la, aquesta actualització es realitza de forma similar als casos anteriors.

**Moure bloc de cel·les**: fem el mateix que al moure una cel·la, però només actualitzem les referències en el cas de les referències per bloc si aquestes estan completament dins del bloc mogut, és a dir si per exemple tenim una referència d’aquest tipus $Full 0-A1:A5 i el bloc mogut comença en la cel·la A1 o anterior i no acaba en la cel·la A5 o posterior, les referències es mantenen.

**Ordenar bloc de cel·les**:

**Cercar**: cerquem dins el full totes les cel·les i emmagatzemem en un ArrayList les posicions que coincideixen amb el valor donat, finalment retornem l’ArrayList.

**Cercar i reemplaçar**: iterem sobre totes les cel·les dins el full i modifiquem el contingut de totes les cel·les que contenen el valor (no el contingut) igual al valor donat.