

Laboratorio 6

Ronald Pardo | 202111309 | r.diazp@uniandes.edu.co | ISIS 1225 (4)

Paso 1

INDICE	MAPTYPE	LOADFACTOR
catalog['artists map'] : TAD MAP(ConstituentID -> DisplayName, Gender, BeginDate, ArtworkNumber, MediumNumber, EndDate, Nationality, Mediums : TAD MAP(), TopMedium , TopMedium artworks)	Linear probing	0.5
catalog['artworks map'] : TAD MAP(ObjectID --> ConstituentID, DateAcquired, CreditLine, Title, Artists : TAD LIST(DisplayName), Classification, Medium, Dimensions, Date)	Linear probing	0.5
catalog['artists with ids'] : TAD MAP(ConstituentID : DisplayName)	Linear probing	0.5
catalog['ids with artists'] : TAD MAP(DisplayName : ConstituentID)	Linear probing	0.5

La razon por la cual decidí usar Linear Probing en todas los indices es porque mi maquina --no tengo pareja-- tiene 8gb de ram y usualmente el tipo de colisiones por Separate Chaining (que denotare por las siglas SC) consumen mucha memoria y no resultan ser eficiente en sus busquedas. Debido a que en las colisiones tipo SC hacen uso de Linked Lists para almacenar buckets lo que las hace dependientes de su funcion de Hash y los pointers que contiene cada elemento en el map agregan memoria, asimismo, el hecho de que se usen Linked Lists para los buckets implica que en ciertos casos la memoria se encuentre en una distancia no cercana entre los objetos, lo cual provoca que hacer una instruccion de busqueda tome mas tiempo en ciertos casos. Adicionalmente me tome el tiempo de medir la cantidad de datos presentes en cada archivo por medio de pruebas de busquedas con la libreria TAD y logre descubrir el numero aproximado de elementos que espero tener en cada indice, por lo que probablemente no ocurrira algun re-hash durante la carga de datos.