

ESTRUCTURAS DE DATOS Y ALGORITMOS

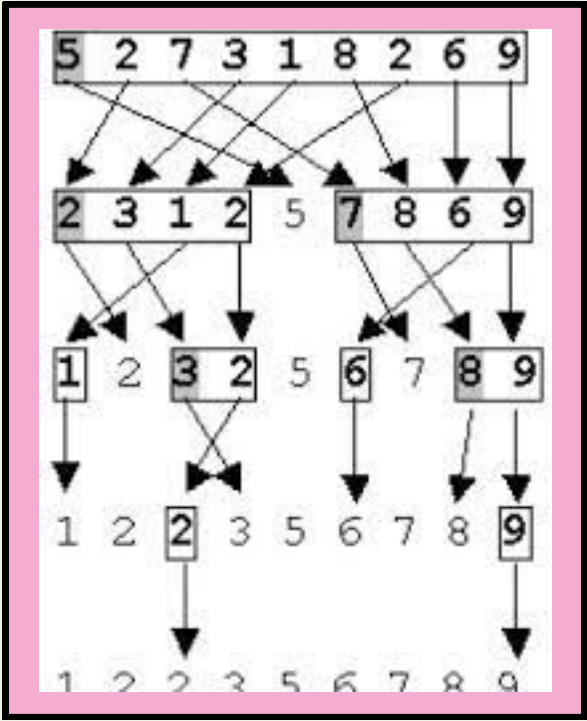
INTERNATIONAL SEAS



POR MEDIO DE ESTRUCTURAS DE DATOS, LOGRAMOS RESOLVER DIVERSOS RETOS Y NECESIDADES PARA LA COMPAÑÍA NAVIERA **"INTERNATIONAL SEAS"** EN EL CANAL DE SUEZ. LAS ESTRUCTURAS DE DATOS IMPLEMENTADAS FUERON LAS SIGUIENTES:

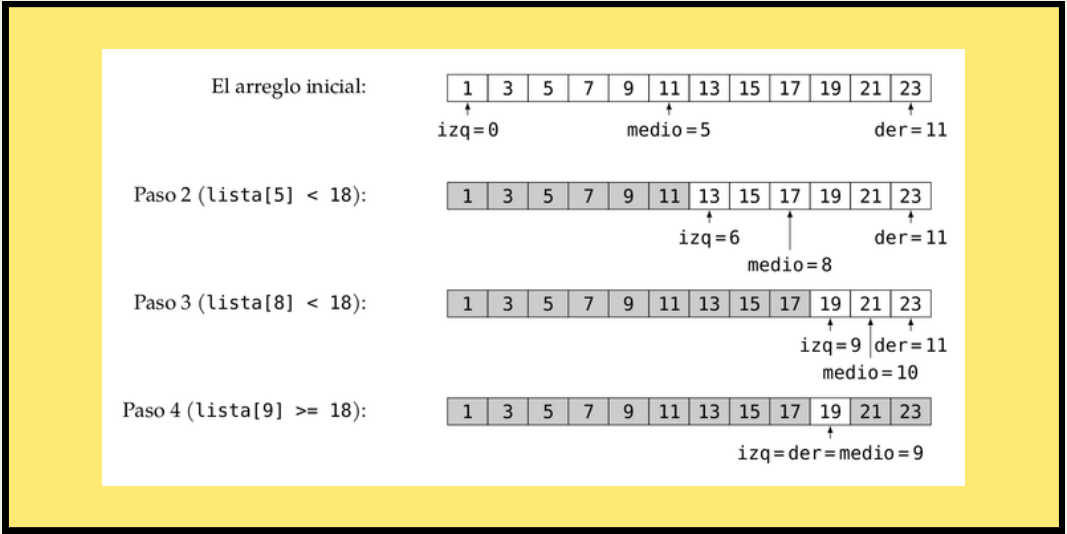
ALGORITMOS DE BÚSQUEDA Y ORDENAMIENTO

NUESTRA NECESIDAD A CUBRIR ERA QUE LA EMPRESA NECESITA QUE SE AUTOMATICE LA BUSQUEDA DE TODS SUS BARCOS, CUYA BASE DE DATOS CONTIENE LA FECHA, LA HORA, Y EL PUNTO DE ENTRADA Y UNA CLAVE DIFERENTE PARA CADA BARCO. LA CUAL EL USUARIO QUE ENCARGADO DE MONITOREAR LA BASE DE DATOS DE LOS BARCOS AL INTRODUCIR LOS 3 PRIMEROS CARACTERES DE LA CLAVE SE MOSTRARIAN TODOS LOS BARCOS EN ORDENADOS POR DIA Y HORA PERO SOLO LOS QUE ESPECIFICA EL USARIO.



PARA LA SOLUCIÓN DE NUESTRO PROBLEMA EMPEZAMOS POR ORDENAR NUESTROS DATOS Y EXISTEN VARIOS TIPOS O MÉTODOS DE ORDENAMIENTO (BUBBLE, SELECTION, INSERTION, MERGE Y QUICKSORT) SOLO HABLARE DEL QUE NOSOTROS USAMOS Y QUE CREEMEOS ES MAS EFICIENTE PARA ESTE PROBLEMA. EL CUAL CONSIDERO QUE EL MAS EFICIENTE HABLANDO DE QUE NUESTRA BASE DE DATOS ES EXTENSA EL MEJOR METODO PARA ORDENARLOS ES EL METODO QUICKSORT YA QUE DIVIDE EL ARREGLO EN DOS SUBARREGLOS PERO A DIFERENCIA EL MERGE ESTE TOMA UN PIVOTE Y RECORRE EL ARREGLO DE MANERA QUE EN UN SUBARREGLO QUEDAN LOS NUMERO MENORES AL PIVOTE Y DEL OTRO LOS MAYORES Y COMIENZA A ORDENAR POR SEPARADO CADA SUBARREGLO PARA DESPUÉS JUNTARLOS YA ORDENADOS.

DE AHI PODEMOS CONTINUAR AHORA SI CON LA BUSQUEDA EN LA BASE DE DATOS, EXISTE LA SECUENCIAL Y LA BINARIA, POR SU PUESTO QUE YA TENIAMOS NUESTROS DATOS ORDENADOS, YA CON NUESTROS DATOS ORDENADOS LA MEJOR IMPLEMENTACIÓN DE BÚSQUEDA PENSAMOS QUE ES LA BINARIA YA QUE ES LA QUE USA LA TECNICA CONOCIDA COMO DIVIDE Y VENCERAS, YA QUE DIVIDE EL ARREGLO JUSTO A LA MITAD Y COMIENZA A BUSCAR LOS DATOS , PORQUE AL SER UNA BASE DE DATOS DEMASIADO LARGA LA SECUENCIAL PODRIA LLEGAR A TARDAR DEMACIADO EN BUSCAR UN DATO DE SER EL EL PENULTIMO O HASTA EL ULTIMO EN UN ARREGLO DE 1000 DATOS. HACIENDO ASI LA BUSQUEDA BINARIA MAS EFICIENTE PARA ESTE CASO, PUEDE SER COMPLEJA DE ENTENDER O PLANTEAR PERO ES EFICIENTE.



COMPLEJIDAD DEL METODO DE ORDENAMIENTO QUICKSORT: $O(N^2)$

COMPLEJIDAD DE LA BUSQUEDA BINARIA: $O(1)$

OTROS USOS
DEL PERIÓDICO

EN EL CANAL DE SUEZ CRUZAN ENTRE 49-97 BARCOS DIARIAMENTE. CADA BARCO TENIA UNA FECHA, HORA, PUNTO DE ENTRADA E IDENTIFICADOR. SE NECESITABA ORDENAR LAS ENTRADAS MES POR MES Y SI ENTRARON POR EL MAR ROJO O MEDITERRANEO

ESTRUCTURAS LINEALES

PARA ESTE RETO SE IMPLEMENTO PRINCIPALMENTE LISTAS DOBLEMENTE ENLAZADAS.

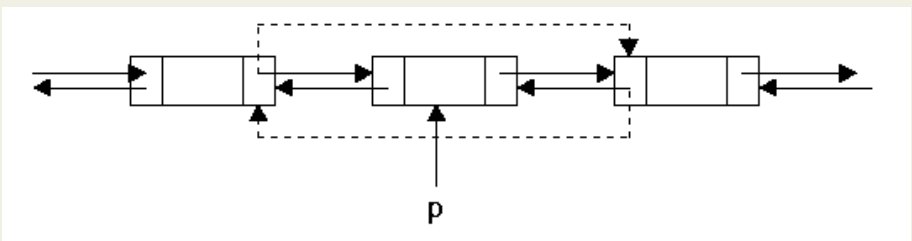
EXISTEN DIFERENTES TIPOS DE ESTRUCTURAS LINEALES COMO:

- LISTAS ENLAZADAS
- FILAS
- PILAS

TODAS ESTAS SE CONSTITUYEN POR UNA SECUENCIA DE DATOS.

PARA RESOLVER ESTE PROBLEMA SE IMPLEMENTARON PRINCIPALMENTE LISTAS DOBLEMENTE ENLAZADAS.

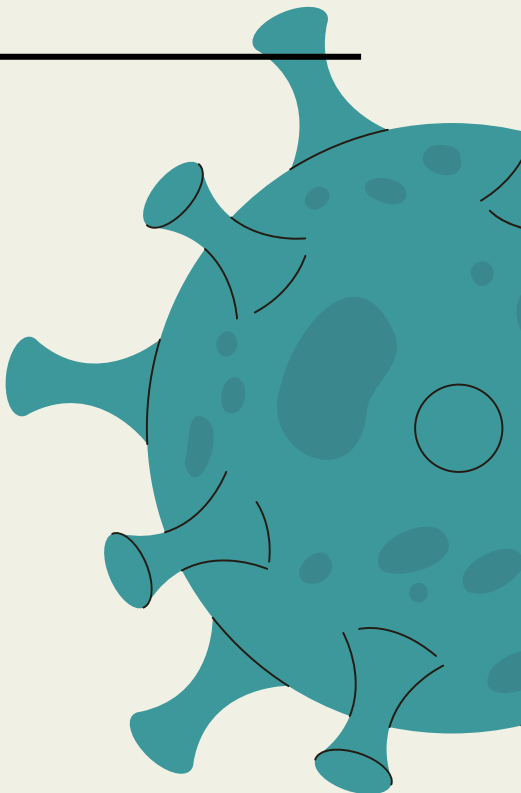
ESTAS LISTAS PERMITEN MOVERSE HACIA DELANTE O HACIA ATRÁS. ESTAS ESTÁN CONECTADAS POR NODOS, IZQUIERDOS Y DERECHOS, QUE TE PERMITEN MOVERTE HACIA DELANTE Y ATRÁS RESPECTIVAMENTE.



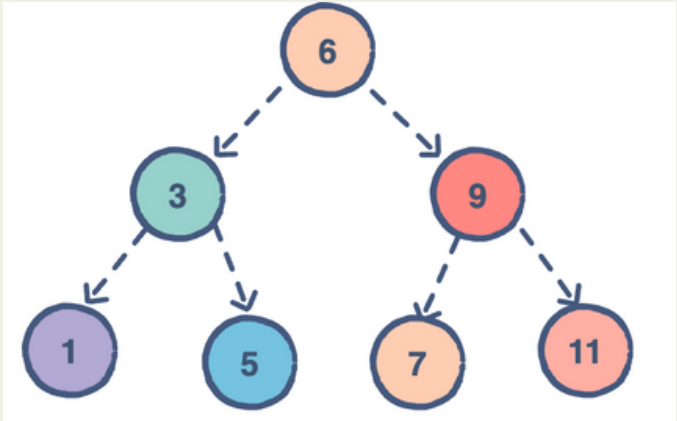
SU COMPLEJIDAD PROMEDIO ES DE $O(N)$

ARBOLES BINARIOS

A CAUSA DE LA PANDEMIA DE COVID-19 LAS OPERACIONES EN LOS PUERTOS SE REDUJERON. AL PRINCIPIO INTERNATIONAL SEAS ORDENABA SUS PEDIDOS POR MEDIO DE LISTAS ORDENADAS PERO PRONTO LOS PEDIDOS SE VIERON SUPERADOS Y SE NECESITARON IMPLEMENTAR ARBOLES BINARIOS DE BUSQUEDA Y SE TENÍA QUE CALCULAR EL NÚMERO MENOR DE COMPARACIONES DE PEDIDOS POSIBLES



LOS ÁRBOLES SON UNA ESTRUCTURA NO LINEAL. LA CUAL ES UN CONJUNTO DE ELEMENTOS POR NODOS, HAY UN NODO PADRE Y SE DERIVAN NODOS HIJOS, CON UN MÁXIMO DE 2 NODOS. UN ÁRBOL BINARIO DE BÚSQUEDA SE DIFERENCIA PORQUE TIENEN UN VALOR ASOCIADO CADA NODO QUE FACILITA LA BÚSQUEDA, AGREGAR O REMOVER SU CONTENIDO.



SU COMPLEJIDAD PROMEDIO ES DE $O(\log(N))$

SUS CARACTERÍSTICAS SON:

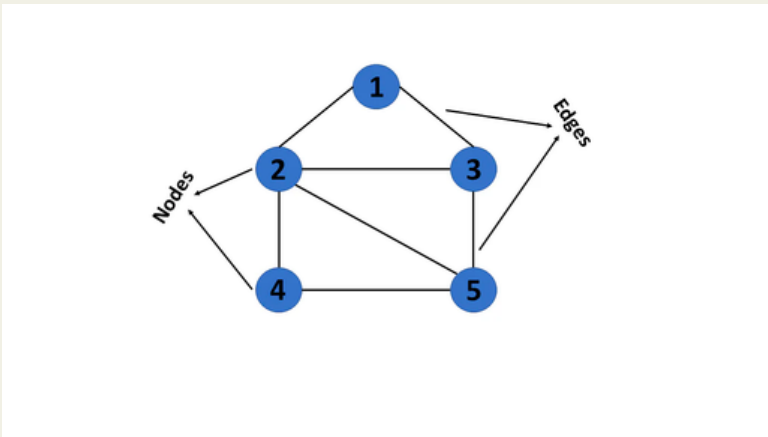
- EL SUBÁRBOL IZQUIERDO SIEMPRE TIENE QUE VALER MENOS QUE EL NODO PADRE
- EL SUBÁRBOL DERECHO SIEMPRE TIENE QUE VALER MÁS QUE EL NODO PADRE.
- LOS NODOS “HIJOS” TIENEN QUE SER ARBOLES BINARIOS.

GRAFOS

UN GRAFO ES UNA ESTRUCTURA NO LINEAR, QUE CONSISTE EN VÉRTICES Y BORDES (EDGES). LOS VÉRTICES SON NODOS Y LOS BORDES SON LÍNEAS QUE CONECTAN A LOS VÉRTICES.

TIENEN MUCHOS USOS EN LA VIDA REAL POR EJEMPLO: FACEBOOK, CADA USUARIO ES UN NODO QUE CONTIENE INFORMACIÓN Y SE CONECTAN POR MEDIO DE “EDGES”

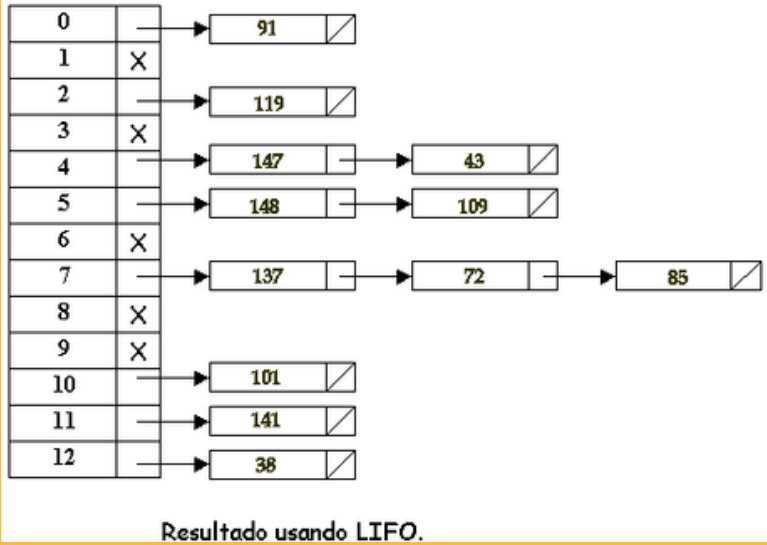
INTERNATIONAL SEAS, DETERMINO QUE CADA EMBARCACION TENDRA UN NUMERO MAXIMO DE PUERTOS QUE PODRA ATRACAR. SOLO PUEDEN EMBARCAR EN 2 PUERTOS Y CADA QUE LLEGAN A UN PUERTO ESTE NÚMERO DISMINUYE. POR MEDIO DE GRAFOS SE DETERMINÓ UNA RUTA PARA SABER A QUÉ PUERTOS UNA EMBARCACIÓN PUEDE LLEGAR Y CUÁLES NO.



SU COMPLEJIDAD PROMEDIO ES DE $O(N)$

HASHING

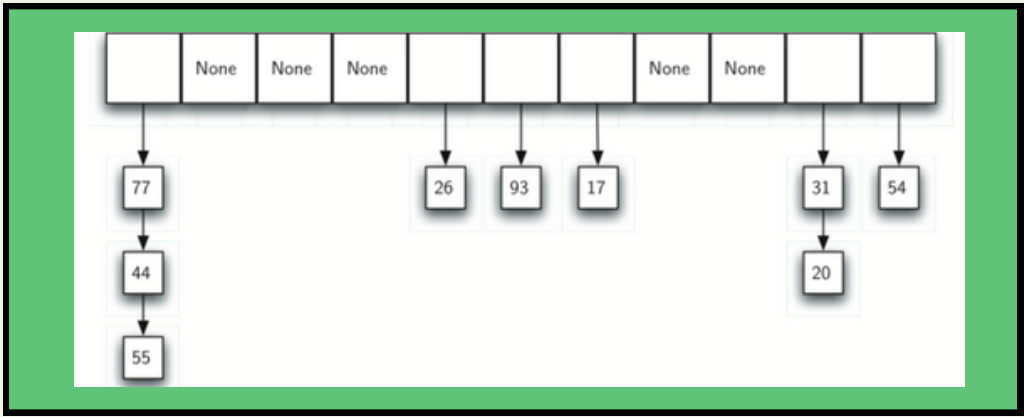
COMO EN CUALQUIER EMPRESA, EN LA COMPAÑÍA NAVIERA "INTERNATIONAL SEAS, LTD." CADA EMPLEADO TIENE UNA DESCRIPCIÓN DE SU PUESTO DE TRABAJO. UNA DESCRIPCIÓN CONTIENE UNOS POCOS PÁRRAFOS QUE DESCRIBE LAS RESPONSABILIDAD DEL PUESTO. ESTA, JUNTO CON OTROS FACTORES, COMO LA ANTIGÜEDAD, SE UTILIZA PARA DETERMINAR SU SALARIO.



PARA ESTE PROBLEMA TENEMOS QUE CALCULAR EL SALARIO DE CADA EMPLEADO DE ACUERDO A SU PUESTO LAS CUALES SON LAS PALABRAS QUE SALEN EN NUESTRA BASE DE DATOS Y LAS QUE NO SE ENCUETREN SERIAN 0

AHORA PARA LA SOLUCION UTILIZAREMOS EL METODOO TECNICA HASHING QUE NOS AYUDARA A DISEÑAR UNA

BUENA FUNCIÓN DE HASHING PARA DISTRIBUIR A LOS ELEMENTOS LO MÁS ALEATORIAMENTE POSIBLE EN LA TABLA, ASI COMO APLICAR UNA ESTRATEGIA EFICIENTE PARA CONTROLAR LAS COLISIONES. EXISTEN VARIOS METODOS DE LA TECNICA HASHING PARA EL TRATAMIENTO DE ESTAS COLSIONES. EL QUE USAMOS Y QUE SON PARECIO UNA MEJOR OPCION. QUE FUE EL METODO LINEAL, NOS HIZO MAS SENCILLO ESE METODO, AUNQUE ESTA EL CUADRATICO TAMBIEN PERO EL PROBLEMA DE ESE ARREGLO ES QUE EN EL CASO DE QUE NUESTRA BASE DE DATOS SEA DEMASIADO GRANDE EL METODO CUADRATICO PUEDE QUE HAYA ESPACIOS SIN VISITAR (SIN CONSIDERAR) EN CAMBIO EL LINEAL CUANDO DETECTA LA COLISION ES CUANO RECORRE EL ARREGLO DE MANERA SECUENCIAL ASEGURANDO QUE NO HABRA UN ESPACIO DEL ARREGLO SIN VISITAR (BASE DE DATOS).



REFERENCIAS:

FERNÁNDEZ J. (SIN FECHA). LISTAS DOBLEMENTE ENLAZADAS. UNIVERSIDAD DE GRANADA. [HTTPS://CCIA.UGR.ES/~JFV/ED1/TEDI/CDROM/DOCS/LDOBLE.HTML](https://ccia.ugr.es/~jfv/ED1/TEDI/CDROM/DOCS/LDOBLE.HTML)

GEEKS FOR GEEKS. (2022). *BINARY SEARCH TREE*. GEEKS FOR GEEKS. [HTTPS://PRACTICE.GEEKSFORGEEKS.ORG/BATCH/CPP-FOUNDATION/TRACK/FOUNDATION-CPP-OOPS/VIDEO/ODY1](https://practice.geeksforgeeks.org/batch/cpp-foundation/track/foundation-cpp-oops/video/ODY1)

GEEKS FOR GEEKS. (2022). INTRODUCTION TO GRAPHS. GEEKS FOR GEEKS. [HTTPS://WWW.GEEKSFORGEEKS.ORG/INTRODUCTION-TO-GRAPHS-DATA-STRUCTURE-AND-ALGORITHM-TUTORIALS/](https://www.geeksforgeeks.org/introduction-to-graphs-data-structure-and-algorithm-tutorials/)