ONLINE AUCTION PLATFORM

ONLINE AUCTION PLATFORM IMPLEMENTED USING A HYBRID DATA STRCTURE FOR EFFICIENT AUCTION MANAGEMENT. COVERS ADDING ITEMS, PLACING BIDS, AND MANAGING THE AUCTION PROCESS.

CB.EN.U4CSE21227 JASWANTH P
CB.EN.U4CSE21215 MAKARAND

Objective and Significance

OBJECTIVES

- Auction item management: To provides a structured approach for adding, searching, and deleting auction items, enabling effective organization and tracking of items.
- Bid placement and tracking: To allow users to place bids on auction items and maintains bid histories, ensuring transparency and facilitating the monitoring of bidding activities.
- Auction control and reporting: To enable the initiation and termination of auctions for specific items, providing information on the highest bid and bid details for effective decision-making.

SIGNIFICANCE

- User-friendly interface: Uses a menu-driven interface, making it user-friendly and accessible to participants, enabling them to interact with the auction platform effortlessly.
- Data management efficiency: Utilizes data structures - hash tables and heaps to efficiently store and retrieve auction items and bids, optimizing performance and resource utilization.
- Enhanced transparency and fairness: Promotes transparency in the auction process by displaying bid histories, current highest bids, and detailed item information, fostering trust and ensuring fair competition among participants.

Hash Table and Heap Implementation

01

Hash Table Implementation

The hash table is used to store and retrieve AuctionItems in the HybridAuctionPlatform. It enables efficient lookup and deletion of items based on their unique item_id.

02

Heap Implementation

The heap is used to store and manage bids in the MaxHeap class within the HybridAuctionPlatform. It allows efficient insertion and retrieval of the highest bid for each item.

03

Integration and Interplay

By integrating these two data structures and effectively managing their interplay, we have created a powerful system that results in faster and more efficient deliveries.

04

Design Choices and Trade-Offs

We made deliberate design choices to balance efficiency with the complexity of the system. By making thoughtful trade-offs, we were able to create a system that is both effective and manageable.

Practical Applications

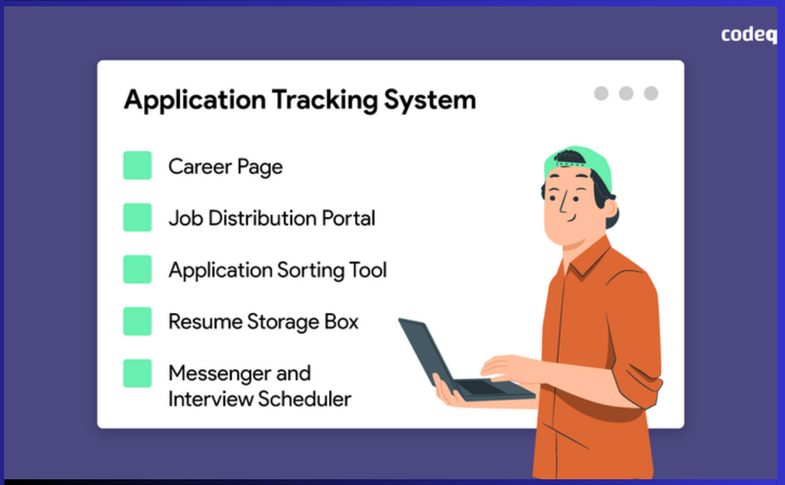
Crowd Funding

The crowdfunding platform incorporates an online auction application, enabling users to participate in auctions for various items and services, fostering engagement and financial support for projects through competitive bidding.

Social Media Influencer Ranking

Algorithmic approach to assess and rank influencers based on engagement, reach, content quality, and audience demographics.

Job Application Tracking



Software to streamline and automate the process of managing job applications, including resume screening, applicant tracking, and candidate communication.

Performance Analysis

Time Complexity

insert() - O(1)
place_bid() - O(1)
end_auction() - O(1)

display_items() - O(n)(where n is number of items)

print_bids() - O(nlogn) (where n is number of bids)

Space Complexity

Items - O(n)
Bids - O(n)

Performance Comparison

The hybrid data structure combines a hash table for efficient item management and a max heap for storing bids. It offers fast insertion, deletion, and searching through the hash table, while maintaining bids in the max heap.

Discussion

1 Practicality and Effectiveness:

- Practicality: It allows users to add items to the auction, place bids, end auctions, view items on auction, and print bid history for an item. The menu-based user interface makes it easy for users to interact with the platform.

- Data Structure Efficiency: A hybrid data structure is used, combining a hash table and a max heap. This combination allows for efficient item retrieval based on item ID using the hash table, while the max heap is used to efficiently retrieve the highest bid for an item. This hybrid approach optimizes the performance of key operations such as placing bids and retrieving item information.

- Collision Handling: Collisions in the hash table are handled by using separate chaining. Each index in the hash table contains a list of item IDs, allowing multiple items to be stored at the same index. This ensures that items with different IDs but the same hash value can coexist without conflicts.

2 Limitations, Challenges, Future Improvements:

While the implemented hybrid data structure demonstrates practicality and effectiveness, there are certain limitations and challenges to consider:

1)Fixed Size
2)Lack of Data Persistence
3)Scalability
4)Error Handling and Validation

Conclusion

- The code implements an auction platform using a hybrid data structure approach, combining a hash table and a max heap.
- The HybridAuctionPlatform class is the main class that manages the auction platform.
- It contains a nested MaxHeap class, which is used to store bids for each item.
- The hash table is implemented using a list of lists (self.table) to handle collisions.
- The code provides methods to add an item to the auction, search for an item, delete an item, place a bid on an item, get a list of all items, display items on auction, print bid history for an item, and end an auction for an item.
- The display_menu function displays a menu with options for the user to interact with the auction platform.
- The main function initializes the HybridAuctionPlatform object and handles user input to perform various actions in the auction.
- Overall, the code provides a functional auction platform with basic operations such as adding items, placing bids, and ending auctions, along with displaying item information and bid history.

REFERENCES

Books:

• [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. , Introduction to Algorithms.
•[2] Narsimha Karumanchi, Data Structures and Algorithms Made Easy

Websites:

• [1] N.E.Goller, “Hybrid Data Structures Defined by Indirection” [Online] Available URL: https://academic.oup.com/comjnl/article/28/1/44/468048
• [2] “Big O Analysis and Hybrid Data Structures” [Online]. Available: URL: https://docs.onenetwork.com/NeoHelp/devnet/big-o-analysisand-hybrid-data-structures-79141067.html
•[3] Dario Radecic, “Data Structures and Algorithms With Python” IEEE Explore. [Online] Available URL: https://towardsdatascience.com/datastructures-and-algorithms-with-python-learn-stacks-queues-and-deques-in-10-minutes-e7c6a2a1c5d5
•[4] JavaTpoint, “Data Structures and Algorithms in Python” . [Online] Available URL: https://www.javatpoint.com/data-structures-and-algorithms-inpython-set-1