



مقدمه

در بسیاری از سیستم های واقعی مانند سیستم های ثبت و قایع، پرونده های پزشکی، لاغهای نرم افزاری و داده های مالی، اطلاعات به صورت رویدادهایی وابسته به زمان ذخیره می شوند. هدف این پژوهه، طراحی و پیاده سازی یک سیستم فهرست بندی زمانی مبتنی بر درخت جستجوی دودویی است تا دانشجویان کاربرد عملی BST را فرآنرا از مثال های ساده درک کنند.

تعريف مسئله

یک سیستم مدیریت رویداد طراحی کنید که هر رویداد شامل اطلاعات زیر باشد:

- شناسه رویداد (EventID)
- زمان وقوع (Timestamp – عدد صحیح)
- دسته بندی رویداد (Category)
- توضیح کوتاه (Description)

درخت جستجوی دودویی باید بر اساس **Timestamp** مرتب سازی شود.

ساختار داده ها

ساختار Event

```
class Event {  
int id;  
int timestamp;  
string category;  
string description;  
};
```

ساختار Node

```
class Node {  
Event data;  
Node* left;  
Node* right;  
};
```

بخش اول: پیاده‌سازی BST (الزامی)

استفاده از ساختارهای آماده کتابخانه‌ای (مانند map یا set) مجاز نیست. موارد زیر باید پیاده‌سازی شوند:

- درج (Insert)
- جستجو (Search)
- حذف (Delete)
- پیمایش inorder

بخش دوم: پرس‌وچهای زمانی

۱. جستجوی بازه‌ای

تابعی پیاده‌سازی کنید که تمام رویدادهای بین دو زمان مشخص را بازیابی کند:

getEventsBetween(t1, t2)

۲. نزدیک‌ترین رویداد زمانی

برای یک زمان داده شده، رویدادی را بیابید که از نظر زمانی کمترین فاصله را دارد:

findClosestEvent(t)

۳. شمارش دسته‌بندی‌ها در بازه زمانی

تعداد رویدادهای هر دسته را در یک بازه زمانی مشخص محاسبه کنید:

countCategories(t1, t2)

بخش سوم: تحلیل عملکرد

برنامه شما باید اطلاعات تحلیلی زیر را گزارش دهد:

- ارتفاع درخت
- تعداد مقایسه‌ها برای هر پرس‌وچو
- میانگین عمق گره‌ها

بخش چهارم: واسط کاربری برنامه

```
Insert event
Delete event
Search by timestamp
Show events in time range
Find closest event
Show tree statistics
Exit
```