



پروژه دوم ساختمان داده

پاییز 1404

دکتر کوهزادی و دکتر قاسمی

## سیستم مدیریت تسک

در این پروژه از شما خواسته می شود که سیستمی برای مدیریت تسک ها پیاده سازی کنید.

هر تسک شامل موارد زیر است:

- 1- id
- 2- start time
- 3- end time
- 4- value

از شما میخواهیم که با استفاده از دیتا استرکت های مختلفی که هر یک برای ما یک مزیت خاص دارد سیستم مد نظر را با ویژگی های خواسته شده پیاده سازی کنید.

1- افزودن تسک جدید: با وارد کردن کامند `InsertTask` باید یک تسک جدید اضافه کنید.

`InsertTask id startTime endTime value`

به عنوان مثال:

InsertTask 1 10 15 30

که در اینجا به عنوان مثال یک تاسک با  
 $\text{id} = 1$  ,  $\text{startTime} = 10$  ,  $\text{endTime} = 15$   $\text{value} = 30$   
 ایجاد کرده ایم.

**2- حذف کردن تاسک :** با دستور `DeleteTask id` تاسک را باید بتوانیم حذف کنیم.  
 به عنوان مثال `deleteTask 2` را باید حذف کند.

**3- آپدیت کردن تاسک :** باید بتوان با استفاده از دستور `updateTask` باید بتوان تاسک را آپدیت کنیم.

UpdateTask id startTime endTime value

به عنوان مثال :

UpdateTask 1 12 17 25

در این حالت تاسک با شماره دارای مقادیر جدید زیر می شود:

`startTime = 12` , `endTime = 17` , `value = 25`

**4- جست و جوی تاسک بر اساس id :** با استفاده از دستور `QueryTaskId id` باید تمامی فیلد های تاسک با `id` را پیدا خروجی دهیم: به عنوان مثال با دستور `1` `startTime` , `endTime` , `value` `QueryTaskId` را برای تاسک با `id = 1` را خروجی دهد.

**5- جست و جو تاسک براساس بازه:** با استفاده از دستور `QueryTaskSum id1 id2` باید جمع `value` تاسک هایی که رنج `id` آنها در این بازه قرار دارند را باید خروجی دهد.  
 به عنوان مثال:

QueryTaskSum 1 3

در این دستور باید جمع مقادیر تاسک هایی که `id` آنها 1 و 2 و 3 است را خروجی دهد.

**6- پیدا کردن تداخل ها :** در زمان اضافه کردن تاسک ها باید در صورت تداخل بین تاسک ها اضافه کردن تاسک متوقف شود و خروجی نمایش داده شود. در صورتی که بین بیشتر از دو تاسک تداخل وجود داشته باشد باید تمامی تداخلات نمایش داده شوند.

**7- چاپ کردن درخت ها:** با وارد شدن دستور `PrintTrees` باید تمامی درخت هایی که پیاده سازی کرده اید را چاپ کنید. برای چاپ کردن اینکه نشان داده شود هر `node` شامل چه اطلاعاتی هست و هر `node` به کدام `node` ها متصل هست باید نشان داده شوند.

**نحوه پیاده سازی:**

در اینجا ما میخواهیم که بتوانیم تسك ها را در یک لیست اصلی نگه داریم. ما در اینجا نیاز داریم که خروجی جست و جوی تسك ها بر اساس id در بدترین حالت در چه زمانی انجام می شود. برای این بخش می توانید از B-Tree استفاده کنید.

برای پیاده سازی سایر ویژگی های برنامه نیز می توانید از درخت های interval tree و segment tree استفاده کنید که هر کدام برای پیاده سازی بخشی کاربرد دارند و می توانند استفاده شوند.

**نکات قابل توجه:**

- 1- باید به تمامی بخش های پروژه تسلط کامل داشته باشید.
- 2- می توانید از ابزار های هوش مصنوعی استفاده کنید.
- 3- برای استفاده از درخت های segment tree , B tree و interval tree باشد و از بخش هایی از درخت که در پروژه استفاده کرده اید باید به نحوه عملکرد آن تسلط داشته باشید.
- 4- دستورات و اطلاعات به صورت داینامیک وارد می شوند و برنامه شما باید بتواند هر حالتی را مدیریت کند.

**منابع کمکی:**

- 1- <https://www.geeksforgeeks.org/dsa/introduction-of-b-tree-2/>
- 2- <https://www.geeksforgeeks.org/dsa/segment-tree-data-structure/>
- 3-<https://www.geeksforgeeks.org/dsa/interval-tree/>