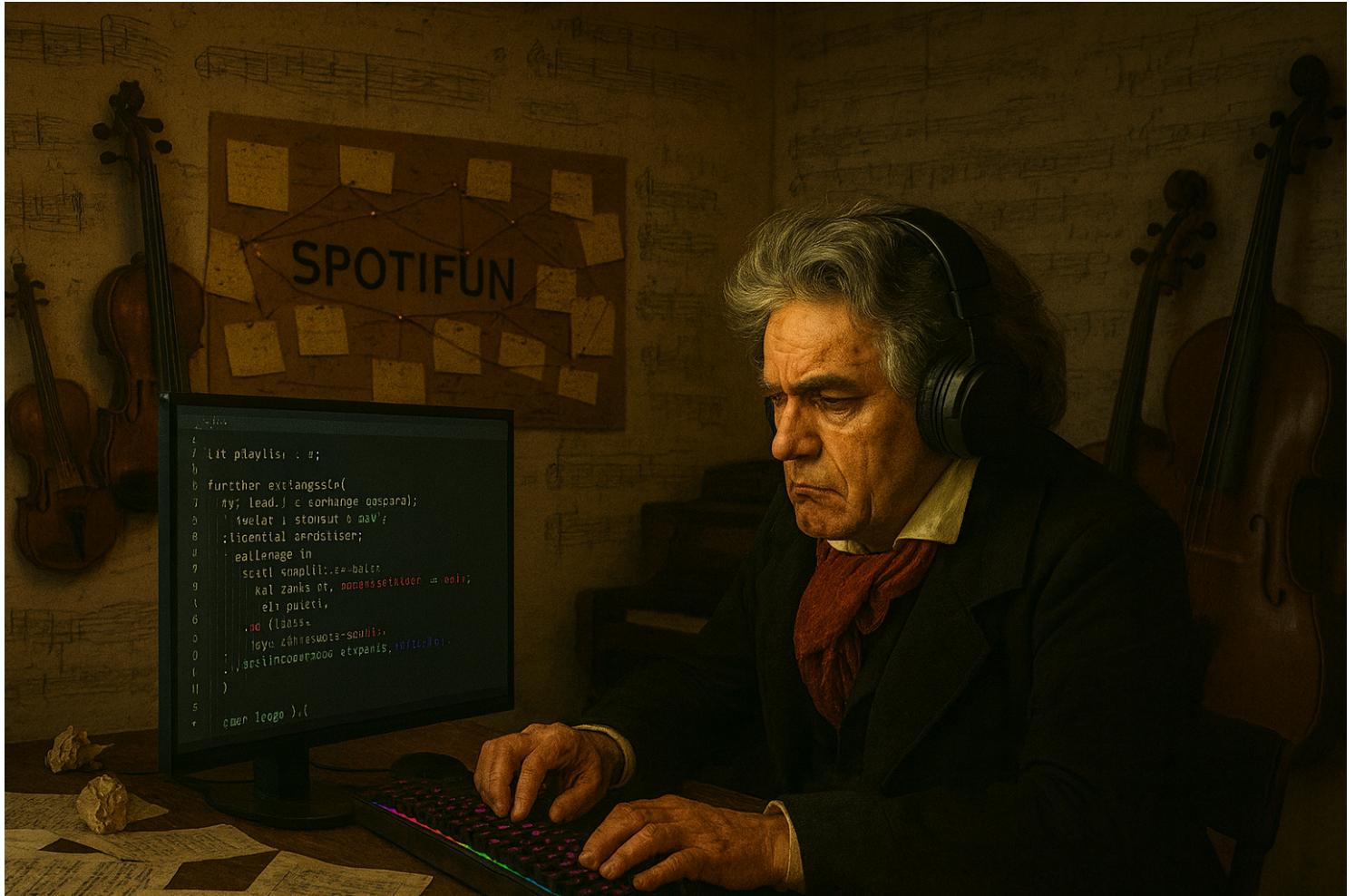


دانشگاه صنعتی خواجه نصیرالدین طوسی

ساختمان داده

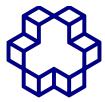
دکتر کوهزادی و دکتر قاسمی



پروژه اول

Spotifun

نیمسال اول 05-04



مقدمه

در دنیای امروز، گوش دادن به موسیقی تنها یک سرگرمی ساده نیست؛ بخشی مهم از زندگی روزمره ماست. برنامه‌ها و سرویس‌های موسیقی با جمع‌آوری آهنگ‌ها، ساخت پلی‌لیست‌های شخصی، نمایش متن موسیقی و ارائه تجربه‌ای روان و لذت‌بخش، توانسته‌اند مخاطبان زیادی را به خود جذب کنند. حالا نوبت شماست که وارد این دنیا شوید!

در این پژوهه قرار است نسخه‌ای قابل توسعه از یک اپلیکیشن مدیریت موسیقی طراحی کنید؛ برنامه‌ای که امکان ثبت خواننده‌ها و آهنگ‌ها، ساخت پلی‌لیست‌های مختلف و جستجوی سریع میان موسیقی‌ها را فراهم می‌کند. هدف اصلی، ساخت یک محیط منظم و کارآمد است که تجربه‌ای شبیه یک برنامه واقعی پخش موسیقی ارائه دهد.

توضیح پژوه

در این پژوهه مجموعه‌ای از خواننده‌ها در اختیار داریم که باید به صورت یک ساختار *Sparse Set* نگهداری شوند. هر خواننده شامل لیستی از آهنگ‌های خود است. فرض کنید تعداد خواننده‌ها کمتر از ۱۰۰ نفر می‌باشد.

هر خواننده دارای اسم و *id* (یکتا) است.

هر آهنگ نیز دارای اسم و *id* (یکتا) و سال انتشار و امتیاز و متن آهنگ است.

در ادامه، مجموعه دستوراتی که باید در این پژوهه پیاده‌سازی شوند معرفی می‌گردند:

اضافه کردن خواننده :

با این دستور می‌توان یک خواننده اضافه کرد.

```
adds artist_id artist_name
```

توجه: در صورتی که *artist_id* تکراری باشد یا بیشتر از ۱۰۰ وارد شود، باید خطای مناسب نمایش داده شود.

حذف خواننده با *id* دلخواه :

با این دستور می‌توان یک خواننده را حذف کرد.

```
del artist_id
```



پیدا کردن یک خواننده با `id` دلخواه :

با این دستور می توان اطلاعات یک خواننده و اسم آهنگهای او را چاپ کرد.

```
finds artist_id
```

چاپ کردن همه خواننده ها :

با این دستور میتوان اطلاعات مربوط به همه ی خواننده ها را چاپ کرد.

```
prints
```

پاک کردن تمام اطلاعات موجود :

با این دستور می توان تمامی اطلاعات موجود را حذف کرد.

```
cls
```

پیدا کردن آهنگ با نام آن و چاپ اطلاعات :

با این دستور میتوان آهنگ مورد نظر را پیدا کرد و اطلاعات مربوط به آن را چاپ کرد.

```
findms music_name
```

اضافه کردن آهنگ :

میتوان با دستور `addms` یک آهنگ را اضافه کرد که شیوه ی صحیح دستور آن به شکل زیر باشد:

```
addms music_name
artist name
year
rating
"your song content"
"your song content"
"your song content"
$end
```



حذف آهنگ (باید متعاقبا از تمام پلی لیست ها و تاریخچه پخش و رتبه بندی آهنگ حذف شود) :

با این دستور می توان این کار را انجام داد.

```
delms artist_id music_id
```

جستجوی کلمه در آهنگ :

با این دستور در متن آهنگ به دنبال کلمه مورد نظر میگردیم و جایگاه آن کلمه را بر میگردانیم.

```
searchw artist_id music_id word
```

تعداد وقوع یک کلمه :

این دستور تعداد وقوع یک کلمه را در متن آهنگ باز میگرداند.

```
countw artist_id music_id word
```

پلی لیست

ساختن یک پلی لیست :

با این دستور یک پلی لیست با نام و id یکتا می سازد.

```
addp playlist_id playlist_name
```

اضافه کردن آهنگ به پلی لیست :

با این دستور یک آهنگ به پلی لیست اضافه میشود.

```
addmp music_id playlist_id
```



پیدا کردن پلی لیست :

با این دستور یک پلی لیست پیدا شده و تمام اطلاعات آن نمایش داده می‌شود. از آنجا که تعداد پلی لیست‌ها معمولاً ثابت و کم‌تغییر است، بهتر است آن‌ها را به شکلی نگهداری کنید که دسترسی به یک پلی لیست خاص سریع و دقیق انجام شود.

`searchp playlist_id`

جستجوی آهنگ :

این دستور یک آهنگ را درون یک پلی لیست جستجو می‌کند.

`searchmp playlist_id music_id`

حذف آهنگ :

این دستور آهنگ را از پلی لیست حذف می‌کند.

`delmp playlist_id music_id`

نشان دادن پلی لیست :

این دستور آهنگ‌های درون پلی لیست را به ترتیب سال مرتب کرده و آنها را چاپ می‌کند.

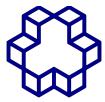
`showp playlist_id`

تاریخچه پخش

پخش آهنگ (اضافه شدن به تاریخچه) :

این دستور، آهنگ مورد نظر را پخش می‌کند.

`playm artist_id music_id`



بازگشت (Undo) و نمایش آخرین آهنگ:

این دستور باید آخرین آهنگ پخش شده را برگرداند.(با اجرای دستور آهنگ ماقبل آخر پخش میشود)

[undo_playm](#)

رتبه‌بندی آهنگ‌ها

نمایش آهنگ با بالاترین امتیاز:

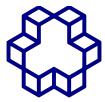
این دستور باید اسم و امتیاز آهنگی که بالاترین رتبه را دارد پیدا کرده و چاپ کند.

[get_max_rated](#)

نمایش آهنگ با پایین‌ترین امتیاز:

این دستور باید اسم و امتیاز آهنگی که پایین‌ترین رتبه را دارد پیدا کرده و چاپ کند.

[get_min_rated](#)



نکات مهم پیاده سازی

۱. ممنوعیت استفاده از ساختمان داده ها و توابع آماده

تمامی ساختمان داده ها و الگوریتم ها باید از پایه پیاده سازی شوند. استفاده از کتابخانه ها یا توابع آماده برای مدیریت داده ها مجاز نیست.

۲. بهینه بودن عملکرد و پیچیدگی ها

هر تابع و عملیاتی که پیاده سازی می کنید باید با بیشترین کارایی اجرا شود. **پیچیدگی زمانی (Time Complexity)** و **پیچیدگی فضایی (Space Complexity)** کد باید بهینه باشد و نمره نهایی پروژه بر اساس این معیارها ارزیابی می شود.

۳. انتخاب ساختار مناسب و استفاده از آرایه تنها در صورت نیاز واقعی

در بخش های مختلف پروژه، انتخاب ساختار داده باید با توجه به هدف و شرایط انجام شود. آرایه می تواند مورد استفاده قرار گیرد، اما تنها زمانی که واقعاً به آن نیاز باشد؛ در غیر این صورت بهتر است از ساختارهای مناسب تر و منعطف تر استفاده شود تا مدیریت داده ها ساده تر و کارآمد تر صورت گیرد.

منابع مفید

- <https://www.geeksforgeeks.org/dsa/sparse-set>
- <https://www.baeldung.com/java-min-max-heap>
- <https://www.geeksforgeeks.org/dsa/priority-queue-set-1-introduction>
- <https://www.geeksforgeeks.org/dsa/suffix-array-set-1-introduction>
- <https://github.com/mniabedii/dsa-masterclass-java>
- <https://www.geeksforgeeks.org/dsa/stack-data-structure>
- <https://www.geeksforgeeks.org/dsa/queue-data-structure>

موفق باشید