موتور جستجو

ساختمان داده‌ها - دکتر رضا رمضانی

دانشگاه اصفهان

زمستان 1403

فهرست

هدف‌ها

هدف‌های اصلی

- شبیه‌سازی عملکرد یک موتور جستجو
- آشنایی با کاربرد ساختمان‌داده‌ی مپ در ذخیره‌سازی و بازیابی داده‌ها
- آشنایی با الگوریتم‌های جستجو
- و ...

هدف‌های جانبی

- رعایت اصول ساختارمندی کد
- رعایت اصول تمیزی کد
- و ...

قابلیت‌ها

- جستجوی بهینه در اسناد متنی بر اساس محتوا
- پشتیبانی از پرس‌وجوهای بولینی
- تشخیص خطاهای متداول

پیش نیازها

- آشنایی با مبانی ساختمان‌داده‌ی مپ
- آشنایی با مبانی الگوریتم و پیاده‌سازی
- آشنایی با فایل‌های متنی و چگونگی کار با آن‌ها

توصیف نیازمندی‌ها

پروژه حاضر با هدف شبیه سازی یک موتور جستجو طراحی شده است که قابلیت پردازش اسناد متنی را داراست. این سیستم امکان جستجوی بهینه با شرایط مختلف شامل کلمات اجباری، اختیاری، و کلمات حذف‌شده را فراهم و نتایج را که شامل اسناد منطبق با شرایط جستجو است خروجی می‌دهد.

-- 50 امتیاز

ساختمان داده‌ها

- نگاشت (Map)
  - پیاده‌سازی ساختمان‌داده‌های مپ **الزامی** بوده و استفاده از پیاده‌سازی‌های آماده‌ی آن در زبان‌های برنامه‌نویسی موجب **کسر نمره‌ی این بخش** خواهد شد.

-- 100 امتیاز

ویژگی‌ها و عملکردها

- توانایی کار با محتوای اسناد متنی و ویرایش آن‌ها
- پشتیبانی از عملگرهای جستجو شامل کلمات مطلوب (+) و کلمات نامطلوب (-)

-- 350 امتیاز

پیاده‌سازی

می‌توان پیاده‌سازی این پروژه را در **سه** مرحله انجام داد:

-- 150 امتیاز

پیش پردازش

-- 50 امتیاز

مرور و پاکسازی اسناد

- اولین قدم بررسی محتوای اولیه اسناد، جداسازی کلمات آن و دور ریختن محتوای اضافی مانند علائم نگارشی، کاراکترهای اضافی و ... به منظور کاهش فضای ذخیره‌سازی مورد نیاز است.

  - توجه کنید که محتوای اصلی اسناد نباید تغییر کند و تغییرات گفته شده صرفا به‌صورت داخلی و در فضای برنامه انجام می‌گیرد.

-- 100 امتیاز

ساخت ایندکس معکوس

- مرحله‌ی بعدی ایجاد ساختاری با قابلیت ذخیره‌سازی و بازیابی بهینه‌ی کلمات است. یک روش برای ایجاد چنین ساختاری ایندکس‌گذاری معکوس نام دارد. ساختار ایجاد شده را به‌دلیل تفاوتی که با ایندکس‌گذاری‌های مرسوم دارد ایندکس معکوس (Inverted Index) می‌نامند.

-- 150 امتیاز

پردازش پرس‌وجوها

- گام نهایی شامل بررسی و پردازش پرس‌وجوهای کاربر با استفاده از ساختار ایجاد شده در مراحل قبل است. پرس‌وجوهای کاربر می‌تواند شامل کلماتی باشد که انتظار دارد:
  - حتما در سند نتیجه‌ی جستجو موجود باشد (بدون عملگر)
  - حداقل یکی از آن‌ها (کلمات) در سند نتیجه‌ی جستجو موجود باشد (شامل عملگر + قبل از آن)
  - اصلا در سند نتیجه‌ی جستجو موجود نباشد (شامل عملگر - قبل از آن)

  - توصیه می‌شود ابتدا کلمات مربوط به هریک از این سه دسته را جداسازی و اسناد مربوط به هرکدام را استخراج کنید؛ سپس با اجرا عملیات منطقی بر اساس جبر مجموعه‌ها، خروجی نهایی را محاسبه و اعلام کنید.

-- 50 امتیاز

ساختار و مدل‌های داده

- رعایت اصول طراحی معماری و سازماندهی کد از جمله اصول SOLID برای **توسعه‌پذیری** و **بهبود کیفیت طراحی** ضروری است.

-- 50 امتیاز

کدنویسی تمیز

رعایت اصول کدنویسی تمیز برای **فهم‌پذیری** و **بهبود کیفیت کد** ضروری است.

-- 50 امتیاز

پیچیدگی و بهینه‌سازی

- پیچیدگی **زمانی** و **فضایی** الگوریتم خود را در **بدترین حالت** محاسبه و **به‌طور مختصر توضیح دهید**.
- بررسی کنید که موتورهای جستجو در دنیای واقعی از چه ساختارهایی برای پاسخگویی بهینه به پرس‌وجوهای کاربران استفاده می‌کنند.

-- 200 امتیاز

ورودی خروجی

ورودی

ورودی شامل n+2 خط است. در خط اول، مسیر نسبی پوشه‌ی حاوی فایل‌ها می‌آید. در خط بعدی عدد n یعنی تعداد پرس‌وجوها دریافت و در هریک از n خط بعدی یک پرس‌وجو دریافت می‌شود. در هر پرس‌وجو کلمات مورد جستجو با یک فاصله از هم جدا شده‌اند و هریک از این کلمات می‌تواند شامل یکی از عملگرهای + یا - به ترتیب برای مشخص کردن کلمات مطلوب یا نامطلوب و یا فاقد عملگر باشد.

$$0 \leq n \leq 10$$

$$0 \leq number\ of\ words\ in\ a\ query \leq 10$$

$$0 < query\ word\ length \leq 20$$

خروجی

خروجی شامل نتایج پرس‌وجوها است که پشت سر هم می‌آیند. نتیجه‌ی هر پرس‌وجو شامل m+1 خط است که m تعداد اسناد **متمایز** منطبق با آن پرس‌وجو می‌باشد؛ در خط اول هر نتیجه، ابتدا عدد m آمده و در m خط بعدی نام اسناد نتیجه به ترتیب **صعودی** می‌آیند. در صورت بروز خطا کافی‌ست متن خطای موردنظر به‌جای نتیجه نمایش داده شود.

مثال:

```
Inp   1
      +Leave -me to +dream
Outp  16
      58775
      58792
      58798
      58800
      58818
      58819
      58830
      58853
      58882
      58917
      59123
      59161
      59227
      59320
      59539
```

با استفاده از این ورودی می‌توان اسنادی را دریافت کرد که حتما شامل کلمه‌ی to و حداقل شامل یکی از کلمات leave و dream هستند، اما اصلا کلمه‌ی me را ندارند.

-- 50 امتیاز

مدیریت خطا

خطاهای ممکن در این سیستم در **دو** بخش کلی طبقه‌بندی می‌شوند:

خطاهای ورودی

- Invalid Input

  مثال:

  +book board +white $classroom -pencil

خطاهای منطقی

- Logical Error

  مثال:

  bench +book +white -bench

  مثال:

  +bench book white -bench

-- 250 امتیاز

امتیازی

-- 60 امتیاز

پشتیبانی از جستجوی فازی

- امکان جستجوی کلمات مشابه با یک یا چند اختلاف، مثلا تغییر در حروف، کم و زیاد شدن تعداد حروف و ...؛

-- 50 امتیاز

پشتیبانی از پرس‌وجوهای الستیک سرچ

- استفاده از ابزار الستیک سرچ elasticsearch برای ذخیره سازی اسناد و بازیابی آن‌ها با استفاده از پرس‌وجوهای آن؛

-- 40 امتیاز

پشتیبانی از جستجوی جملات

- افزودن امکان جستجوی اسناد بر مبنای رخداد جملات؛

-- 30 امتیاز

پشتیبانی از جستجو در انواع مختلف فایل

- امکان جستجو در اسناد از انواع مختلف فایل مانند CSV، JSOT , ..؛

-- 20 امتیاز

پشتیبانی از رابط کاربری گرافیکی

- پیاده‌سازی رابط کاربری گرافیکی به سبک دلخواه برای موتور جستجو؛

-- 50> امتیاز

سایر موارد

- گسترش قابلیت‌های موتور جستجو بسته به خلاقیت و سلیقه‌ی شخصی؛

- به منظور حفظ عملکرد حالت پایه‌ی موتور جستجو هریک از پیاده‌سازیی‌های امتیازی باید با وارد کردن دستور خاصی مثلا Advanced Mode اجرا شوند و در غیر اینصورت برنامه صرفا از ورودی خروجی‌های گفته شده در بخش‌های قبلی پشتیبانی می‌کند. پیروی از این الگو برای قبولی تست‌کیس‌ها ضروری است.
- پیاده‌سازی موارد امتیازی تنها در صورت تکمیل بخش‌های اصلی پروژه می‌تواند امتیاز اضافی به همراه داشته باشد.

نکات تکمیلی

زبان پیاده‌سازی

- زبان‌های برنامه‌نویسی قابل ارزیابی در پروژه شامل C++ و java و C# می‌باشد.
- استفاده از فریم‌ورک زبان‌های برنامه‌نویسی نام‌برده مجاز است.

مهلت تکمیل

- پروژه‌ی جاری در تاریخ 1403/10/09 ساعت 00:00:00 منتشر و حداکثر مهلت اتمام آن تا تاریخ 1403/10/18 ساعت 06:00:00 می‌باشد.

گروهبندی

- پیاده‌سازی و ارزیابی پروژه به‌صورت فردی انجام می‌شود.

بستر پیاده‌سازی

- توسعه و پیاده‌سازی پروژه، در بستر گیت‌هاب انجام می‌گیرد.
- با ورود به لینک اساینمنت پروژه، ریپوزیتوری مخصوص هر فرد به‌صورت خودکار ساخته می‌شود.
- روند توسعه‌ی پروژه در قالب کامیت‌های **متوالی** و **معنادار** روی برنچی به نام Search Engine و تحلیل پیچیدگی آن روی **برنچ دیگری** بنام Analysis انجام می‌شود. توصیه می‌شود پیاده‌سازی بخش‌های امتیازی نیز روی برنچ مجزا انجام گردد.
- ساخت **برنچ‌های متعدد** و رعایت اصول نوشتار صحیح متن کامیت برای توسعه‌ی تمیزتر توصیه می‌شود.

نحوه‌ی ارزیابی

- ارزیابی عملکرد پروژه، به‌صورت تست‌کیسی در بستر گیت‌هاب و همچنین ارائه‌ی حضوری انجام می‌گیرد.
- ارزیابی ابتدایی پروژه به کمک تست‌کیس‌های طبقه‌بندی شده و با ابزار تست گیت‌هاب انجام می‌شود.
- ارزیابی نهایی در قالب ارائه‌های 20 دقیقه‌ای، به‌صورت حضوری و براساس **بارم‌بندی ذکرشده در بخش‌های مختلف همین مستند** انجام می‌شود.
- بخش‌های امتیازی پروژه شامل ارزیابی اولیه **نمی‌شود**.