این پروژه در مورد پیاده‌سازی یک درخت AVL در زبان Rust است که می‌تواند عناصر را به درخت اضافه کند، تعادل را حفظ کند و عناصر را جستجو کند. در زیر خلاصه‌ای از این کد آورده شده است:

**هدف پروژه**

پیاده‌سازی یک درخت AVL که:

1. عناصر را با حفظ ترتیب در درخت قرار دهد.
2. پس از هر افزودن، تعادل درخت را از طریق چرخش‌ها حفظ کند.
3. قابلیت جستجوی سریع عناصر را فراهم کند.

**ساختارهای داده**

1. **Node** : هر گره شامل مقدار، ارتفاع و اشاره‌گر به گره‌های فرعی (چپ و راست) است.
2. **AVLTree** : ساختار درخت که اشاره‌گری به ریشه درخت دارد.

**توابع اصلی**

1. **افزودن عنصر (insert)**
   * عنصر به درخت اضافه می‌شود.
   * ارتفاع هر گره به‌روزرسانی شده و در صورت نیاز عملیات متعادل‌سازی انجام می‌شود.
2. **متعادل‌سازی (balance)**
   * با استفاده از **چرخش چپ** و **چرخش راست**، درخت متعادل می‌شود.
   * چرخش‌ها شامل:
     + چرخش ساده (چپ یا راست)
     + چرخش دوبل (چپ-راست یا راست-چپ)
3. **جستجو (search)**
   * بررسی می‌کند که آیا یک مقدار در درخت وجود دارد یا خیر.
4. **کمکی‌ها**
   * محاسبه ارتفاع گره‌ها.
   * محاسبه فاکتور تعادل برای بررسی نیاز به متعادل‌سازی.

**ویژگی‌های کد**

* از **Option<Box<Node<T>>>**  برای مدیریت اشاره‌گرها استفاده شده است.
* از ویژگی‌های **Clone** و **Ord** برای مدیریت مقادیر گره‌ها بهره برده شده است.
* کد دارای ساختاری خوانا و مقیاس‌پذیر است.