**خلاصه پروژه: پیاده‌سازی MaxHeap در Rust**

این پروژه یک پیاده‌سازی ساده از **MaxHeap** (هیپ بزرگ‌ترین-اول) است که به عنوان یک ساختار داده مبتنی بر **درخت دودویی کامل** طراحی شده است. هدف این پروژه فراهم کردن قابلیت مدیریت مؤثر داده‌ها به گونه‌ای است که همیشه بزرگ‌ترین عنصر در دسترس باشد.

**ویژگی‌های اصلی پروژه**

1. **ساختار داده:**
   * داده‌ها در یک لیست (Vec<i32>) ذخیره می‌شوند.
   * ترتیب درخت دودویی در لیست به‌گونه‌ای حفظ می‌شود که هر گره بزرگ‌تر یا مساوی فرزندان خود باشد.
2. **عملیات اصلی:**
   * **افزودن عنصر (push)**:  
     یک عدد به هیپ اضافه شده و ساختار هیپ بازسازی می‌شود (با استفاده از heapify\_up).
   * **حذف بزرگ‌ترین عنصر (pop)**:  
     عنصر ریشه (بزرگ‌ترین) حذف شده و هیپ بازسازی می‌شود (با استفاده از heapify\_down).
   * **مشاهده بزرگ‌ترین عنصر (peek)**:  
     مقدار ریشه (بزرگ‌ترین عنصر) بدون حذف نمایش داده می‌شود.
3. **بازسازی هیپ:**
   * **heapify\_up**: بازسازی از پایین به بالا برای قرار دادن عنصر جدید در موقعیت صحیح.
   * **heapify\_down**: بازسازی از بالا به پایین پس از حذف بزرگ‌ترین عنصر.

**نتیجه پروژه**

این پروژه با استفاده از MaxHeap موارد زیر را مدیریت می‌کند:

* افزودن داده‌ها به صورت پویا.
* نگه داشتن بزرگ‌ترین عنصر همیشه در صدر (ریشه).
* بازسازی مؤثر ساختار هیپ با پیچیدگی زمانی:
  + O(log n) برای عملیات push و pop.
  + O(1) برای مشاهده بزرگ‌ترین عنصر.