**بزرگ‌ترین مستطیل در ماتریس باینری**

**مقدمه**

مسئله بزرگ‌ترین مستطیل در ماتریس باینری یکی از مسائل چالش‌برانگیز در ساختمان داده و الگوریتم‌ها است. هدف این مسئله، یافتن بزرگ‌ترین مستطیل تشکیل شده از 1‌ها در یک ماتریس باینری است. این پروژه به طراحی و پیاده‌سازی الگوریتمی برای حل این مسئله اختصاص دارد.

**شرح مسئله**

به ما یک ماتریس باینری داده شده است که شامل 0 و 1 است. هدف، یافتن مساحت بزرگ‌ترین مستطیلی است که فقط شامل 1‌ها باشد.

**مثال:**

ورودی:

1 0 1 1

1 1 1 1

0 1 1 0

خروجی:

4

در این مثال، بزرگ‌ترین مستطیل 1‌ها شامل 4 سلول متوالی است.

**روش حل مسئله**

برای حل این مسئله، از ترکیب چندین الگوریتم استفاده می‌کنیم:

1. **اسکن هر ردیف ماتریس**: در هر مرحله، ردیف‌های بالاتر را با ردیف فعلی ترکیب کرده و به یک آرایه هیستوگرام تبدیل می‌کنیم.
2. **استفاده از الگوریتم بزرگ‌ترین مستطیل در هیستوگرام**: از این الگوریتم برای یافتن بزرگ‌ترین مساحت مستطیل در هیستوگرام فعلی استفاده می‌کنیم.

**ورودی‌ها و خروجی‌ها**

**ورودی:**

1. تعداد ردیف‌ها و ستون‌های ماتریس
2. ماتریس باینری

**خروجی:**

* مساحت بزرگ‌ترین مستطیل تشکیل شده از 1‌ها.

**نمونه ورودی و خروجی**

**نمونه 1:**

ورودی:

4 4

1 0 1 0

1 0 1 1

1 1 1 1

0 1 1 0

خروجی:

6

**نمونه 2:**

ورودی:

3 3

1 1 1

1 1 1

0 0 1

خروجی:

6

**مراحل پیاده‌سازی**

1. دریافت ورودی و ذخیره ماتریس باینری.
2. ایجاد آرایه‌ای برای ذخیره هیستوگرام هر ردیف.
3. پیمایش هر ردیف:
   * بروزرسانی هیستوگرام بر اساس ردیف فعلی.
   * محاسبه بزرگ‌ترین مستطیل در هیستوگرام با استفاده از پشته.
4. چاپ مساحت بزرگ‌ترین مستطیل.

**نکات پیاده‌سازی**

1. استفاده از **پشته** برای بهینه‌سازی محاسبه بزرگ‌ترین مستطیل در هیستوگرام.
2. اطمینان از پردازش درست آرایه‌ها در مرزهای ماتریس.
3. مدیریت مقادیر صفر در ماتریس برای جلوگیری از خطا.

**موارد نمره مثبت**

* استفاده از کامنت‌گذاری مناسب برای توضیح کد.
* نمایش گرافیکی ماتریس و بزرگ‌ترین مستطیل (در صورت امکان(
* بهینه‌سازی حافظه با استفاده از آرایه‌های دینامیک.
* تست کد با ماتریس‌های بزرگ و بررسی عملکرد.