

خطوات الحل

1. نبدأ بخطتين: واحد عند البداية والآخر عند النهاية.
2. نحسب الحجم المحتمل.
3. نقارن الحجم مع الحجم الأقصى المسجل.
4. إذا كان ارتفاع الخط على اليسار أقل، نتحرك نحو اليمين، وإذا كان ارتفاع الخط على اليمين أقل، نتحرك نحو اليسار.
5. نكرر هذه العملية حتى نلتقي.

كود الحل:

```
:(def max_area(height
left = 0
right = len(height) - 1
max_volume = 0

:while left < right
# حساب الحجم
([h = min(height[left], height[right]
width = right - left
volume = h * width

# تحديث الحجم الأقصى
(max_volume = max(max_volume, volume

# تحريك المؤشر
:[if height[left] < height[right]
left += 1
:else
right -= 1

return max_volume
```

مثال على الاستخدام

```
[height = [1,8,6,2,5,4,8,3,7
(output = max_area(height
(print(output
```

الناتج سيكون 49

أمثلة

1. المثال الأول:

- المدخلات: `height = [1,8,6,2,5,4,8,3,7]`
- الناتج: `49`

2. المثال الثاني:

- المدخلات: `height = [1,1]`
- الناتج: `1`

الخلاصة

باستخدام هذا الأسلوب، نستطيع تحديد أقصى حجم من الماء الذي يمكن أن يحتفظ به الوعاء باستعمال خوارزمية فعالة تعتمد على مؤشرات مزدوجة، مما يجعل الحل فعالاً من حيث الزمن.