DSA طوابير

قوائم الانتظار

.قائمة الانتظار هي بنية بيانات يمكن أن تحتوي على العديد من العناصر

.فكر في الطابور كأشخاص يقفون في طابور في سوبر ماركت

:FIFO أول شخص يقف في الطابور هو أيضًا أول من يستطيع الدفع ومغادرة السوبر ماركت. تسمى هذه الطريقة في تنظيم العناصر .من يدخل أولاً يخرج أولاً

العمليات الأساسية التي يمكننا إجراؤها على قائمة الانتظار هي

- .الإنكويو: إضافة عنصر جديد إلى قائمة الانتظار
- .إلغاء قائمة الانتظار: إزالة وإرجاع العنصر الأول (الأمامي) من قائمة الانتظار
- .نظرة خاطفة : تُرجع العنصر الأول في قائمة الانتظار
- .هو فارغ: يتحقق مما إذا كانت قائمة الانتظار فارغة
- .الحجم: يبحث عن عدد العناصر في قائمة الانتظار

.جرّب هذه العمليات الأساسية في الرسوم المتحركة لقائمة الانتظار أعلاه

.يمكن تنفيذ قوائم الانتظار باستخدام المصفوفات أو القوائم المرتبطة

يمكن استخدام قوائم الانتظار لتنفيذ جدولة المهام لطابعة مكتبية، أو معالجة الطلبات للتذاكر الإلكترونية، أو لإنشاء خوارزميات للبحث .المتسع أولًا في الرسوم البيانية

. وغالبًا ما يتم ذكر قوائم الانتظار مع الأكوام، وهي بنية بيانات مشابهة موصوفة في

تنفيذ قائمة الانتظار باستخدام المصفوفات

لفهم أفضل لفوائد استخدام المصفوفات أو القوائم المرتبطة لتنفيذ قوائم الانتظار، يجب عليك الاطلاع على ذلك الذي يشرح كيفية تخزين .المصفوفات والقوائم المرتبطة في الذاكرة

:هكذا يبدو الأمر عندما نستخدم مصفوفة كقائمة انتظار

:أسباب تنفيذ قوائم الانتظار باستخدام المصفوفات

- .كفاءة الذاكرة: لا تحتفظ عناصر المصفوفة بعنوان العناصر التالية مثل عقد القائمة المرتبطة
- أسهل في التنفيذ والفهم: يتطلب استخدام المصفوفات لتنفيذ قوائم الانتظار تعليمة برمجية أقل من استخدام القوائم المرتبطة، .ولهذا السبب عادةً ما تكون أسهل في الفهم أيضًا

:أسباب عدم استخدام المصفوفات لتنفيذ قوائم الانتظار

- الحجم الثابت: تشغل المصفوفة جزءًا ثابتًا من الذاكرة. هذا يعني أنها قد تشغل ذاكرة أكثر من اللازم، أو إذا امتلأت المصفوفة، فلا يمكنها استيعاب المزيد من العناصر. ويمكن أن يكون تغيير حجم المصفوفة مكلفًا.
- تكلفة التحويل: تتسبب عملية الإزالة في إزالة العنصر الأول في قائمة الانتظار، ويجب إزاحة العناصر الأخرى لتأخذ مكان العناصر التي تمت إزالتها. هذا غير فعال ويمكن أن يسبب مشاكل، خصوصًا إن كانت قائمة الانتظار طويلة.
- البدائل: تحتوي بعض لغات البرمجة على بنيات بيانات مدمجة محسّنة لعمليات قائمة الانتظار أفضل من استخدام المصفوفات.

ملاحظة: عند استخدام المصفوفات في بايثون في هذا البرنامج التعليمي، نحن نستخدم نوع بيانات "قائمة" بايثون، ولكن في نطاق هذا البرنامج التعليمي يمكن استخدام نوع بيانات "القائمة" بنفس طريقة استخدام المصفوفة. تعرف على المزيد حول قوائم بايثون .

نظرًا لأن قوائم بايثون لديها دعم جيد للوظائف اللازمة لتنفيذ قوائم الانتظار، نبدأ بإنشاء قائمة انتظار والقيام بعمليات قائمة الانتظار ببضعة أسطر فقط:

مثال

بايثون

```
قائمة الانتظار = []

# إنشاء قائمة انتظار
قائمة الانتظار.append('A')
قائمة الانتظار.append('B')
إلحاق قائمة الانتظار('C')
طباعة("قائمة الانتظار: "، قائمة الانتظار)

# إلغاء قائمة الانتظار
عنصر = قائمة الانتظار.pop(0)
طباعة("إلغاء قائمة الانتظار: "، عنصر)

# نظرة خاطفة
العنصر الأمامي = قائمة الانتظار[0]
طباعة("نظرة خاطفة: "، العنصر الأمامي)

# هي فارغة
isEmpty = ليس bool(قائمة الانتظار)
طباعة("isEmpty: "، isEmpty)

# الحجم
طباعة("الحجم: "، len(قائمة الانتظار))
```

لكن لإنشاء بنية بيانات لقوائم الانتظار بشكل صريح، مع العمليات الأساسية، يجب أن ننشئ فئة قائمة انتظار بدلًا من ذلك. تشبه طريقة إنشاء قوائم الانتظار هذه في Python أيضًا طريقة إنشاء قوائم الانتظار في لغات برمجة أخرى مثل C و Java.

مثال

بايثون

```
صنف قائمة انتظار:
    def __init__(self):
        قائمة الانتظار الذاتية = []

    تعريف enqueue(ذاتي، عنصر):
```

```
(عنصر)queue.append(عنصر) إلحاق .queue ذاتي

(ذاتي)تعريف إلغاء قائمة الانتظار :
self.isEmpty() إذا كانت :
"إرجاع "قائمة الانتظار فارغة
self.queue.pop(0) إرجاع

(ذاتي)تعريف نظرة خاطفة :
self.isEmpty() إذا كانت :
"إرجاع "قائمة الانتظار فارغة
self.queue[0] إرجاع

isEmpty(self) تعريف :
len(self.queue) = 0 إرجاع

(ذاتي)تعريف الحجم :
len(self.queue) إرجاع

# إنشاء قائمة انتظار
()قائمة الانتظار = قائمة الانتظار

myQueue.enqueue('A')
myQueue.enqueue('B')
myQueue.enqueue('C')
(myQueue.queue.queue ،" :طباعة("قائمة انتظار

(myQueue.dequeue() ،" :طباعة("إلغاء قائمة الانتظار

(myQueue.peekue() ،" :طباعة("نظرة خاطفة

طباعة("isEmpty: " ،myQueue.isEmpty())

(myQueue.size() ،" :طباعة ("الحجم
```

تنفيذ قائمة الانتظار باستخدام القوائم المرتبطة

:أسباب استخدام القوائم المرتبطة لتنفيذ قوائم الانتظار

- .الحجم الديناميكي: يمكن أن تنمو قائمة الانتظار وتتقلص ديناميكيًا، على عكس المصفوفات
- دون الحاجة إلى إزاحة العناصر الأخرى في (enqueue) عدم التحويل: يمكن إزالة العنصر الأمامي من قائمة الانتظار .الذاكرة

:أسباب عدم استخدام القوائم المرتبطة لتنفيذ قوائم الانتظار

- ذاكرة إضافية: يجب أن يحتوي كل عنصر من عناصر قائمة الانتظار على عنوان العنصر التالي (عقدة القائمة المرتبطة .(التالية
- .سهولة القراءة: قد يكون من الصعب قراءة الرمز وكتابته بالنسبة للبعض لأنه أطول وأكثر تعقيدًا

.هذه هي الطريقة التي يمكن بها تنفيذ قائمة انتظار باستخدام قائمة مرتبطة

مثال

بايثون

```
صنف العقدة:
    تعريف __init__(الذات، البيانات):
        الذات.data = البيانات
        الذات.التالي = لا شيء

صنف قائمة انتظار:
    تعريف __init___(ذاتي):
        الذات.front = لا شيء
        الذات.الخلفية = لا يوجد
        الطول الذاتي = 0

    تعريف enqueue(ذاتي، عنصر):
        عقدة_جديدة = عقدة (عنصر)
        إن كانت الذات.الخلفية لا شيء:
            الذات.الأمامية = الذات.الخلفية = العقدة الجديدة
            الطول الذاتي =+ 1
            إرجاع
        الذات.الخلفية.التالي = العقدة الجديدة
        self.rear = new_node
        الطول =+ 1

    Def dequeue(self):
        إذا كان self.isEmpty():
            "إرجاع "قائمة الانتظار فارغة
        temp = self.front(self.front)
        الذات.front = temp.next
        الطول الذاتي =- 1
        إذا كان self.front لا شيء
            الذات.الخلفية = لا شيء
        إرجاع temp.data

    تعريف نظرة خاطفة (ذاتي):
        إذا كانت الذات.isEmpty():
            "إرجاع "قائمة الانتظار فارغة
        إرجاع self.front.data

    تعريف isEmpty(الذات):
        إرجاع self.length = = 0

    تعريف الحجم(ذاتي):
        إرجاع الطول الذاتي

    تعريف طباعة قائمة الانتظار(ذاتي):
        الذات.المؤقّت = front
        بينما المؤقت:
            طباعة (temp.data، النهاية="")
            temp = temp.next
        طباعة()
```

```
إنشاء قائمة انتظار #
()قائمة الانتظار = قائمة الانتظار

myQueue.enqueue('A')
myQueue.enqueue('B')
myQueue.enqueue('C')
("""=طباعة("قائمة الانتظار: "، النهاية
myQueue.printQueue()

(("إلغاء قائمة الانتظار"طباعة: "، myQueue.dequeue()))

("نظرة خاطفة"طباعة: "، myQueue.peek())

("isEmpty: "، myQueue.isEmpty()))طباعة

("الحجم"طباعة: "، myQueue.size())
```