

d1

✓ Klar

Starta uppgift

Inlämningsdatum 24 nov 2019 av 23.59 **Poäng** 1 **Lämnar in** en filuppladdning
Filtyper py och txt

d1

Mål: läsa Pythons dokumentation, använda moduler, implementera en datatyp, hantera länkade listor.

"Trollkarlen tar ut de tretton spaderna ur leken, håller dem som en kortlek med baksidan upp och lägger ut dem på följande sätt: Översta kortet stoppas underst, nästa kort läggs ut med framsidan upp, nästa kort stoppas underst, nästa kort läggs ut osv. Till publikens häpnad kommer korten upp i ordning ess, tvåa, trea...
Utförande: Man ordnar i hemlighet korten enligt följande:"

Ja, här bryter vi citatet ur Liberg: Trolleri för alla.

I labbuppgiften ingår nämligen att ta reda på kortkonstens hemlighet! Du ska därför göra ett program där man kan simulera korttricket så här:

Vilken ordning ligger korten i?
3 1 4 2 5
De kommer ut i denna ordning: 1 2 3 5 4

I denna labb ska du implementera en kö på två olika sätt. Med den abstrakta datastrukturen kö kan man göra tre saker:

enqueue(x) stoppa in något sist
x = dequeue() plocka ut det som står först
isEmpty() kolla om kön är tom

[_\(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#uppgifter\)_](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#uppgifter) Uppgifter

I första uppgiften ska du skriva en egen klass ArrayQ där du implementerar en kö med hjälp av pythons [array](https://docs.python.org/3/library/array.html) [_\(https://docs.python.org/3/library/array.html\)](https://docs.python.org/3/library/array.html)

[_\(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#uppgifter)

tilpro/blob/master/trollkarlsko/d1.org#egna-experiment)_Egna experiment

Börja med att [importera modulen](https://docs.python.org/3/tutorial/modules.html) [array](https://docs.python.org/3/library/array.html) [_array](https://docs.python.org/3/library/array.html) [_array](https://docs.python.org/3/library/array.html) med `from array import array`. Skriv ett provapå-program där du skapar en array (bestäm dig för en datatyp du vill lagra) och experimentera med array-metoderna

- [append](https://docs.python.org/3/library/array.html#array.array.append) [_append](https://docs.python.org/3/library/array.html#array.array.append)
- [insert](https://docs.python.org/3/library/array.html#array.array.insert) [_insert](https://docs.python.org/3/library/array.html#array.array.insert)
- [remove](https://docs.python.org/3/library/array.html#array.array.remove) [_remove](https://docs.python.org/3/library/array.html#array.array.remove)
- [pop](https://docs.python.org/3/library/array.html#array.array.pop) [_pop](https://docs.python.org/3/library/array.html#array.array.pop)

Rita bilder som illustrerar vad metoderna gör. Vilka vill du använda i din enqueue respektive dequeue?

[_tilpro/blob/master/trollkarlsko/d1.org#arrayq---en-k%C3%B6-med-Pythons-array](https://github.com/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#arrayq---en-k%C3%B6-med-Pythons-array))_ArrayQ - en kö med Pythons array

Nu är du redo att skriva din egen klass ArrayQ.

Attribut En array (och ev andra attribut som du vill ha med). Alla attribut ska vara privata (inledas med understrykningstecken `_`).

Metoder `__init__`, `enqueue`, `dequeue` och `isEmpty` (men inga andra metoder).

[_tilpro/blob/master/trollkarlsko/d1.org#testa-arrayq](https://github.com/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#testa-arrayq))_Testa ArrayQ

Prova din kö med följande testfunktion:

```
def basictest():
    q = ArrayQ()
    q.enqueue(1)
    q.enqueue(2)
    x = q.dequeue()
    y = q.dequeue()
    if (x == 1 and y == 2):
        print("test OK")
    else:
        print("FAILED expected x=1 and y=2 but got x =", x, " y =", y)
```

[_tilpro/blob/master/trollkarlsko/d1.org#skriv-trollkarlsprogrammet](https://github.com/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#skriv-trollkarlsprogrammet))_Skriv Trollkarlsprogrammet

Skriv ett program som simulerar korttricket som står beskrivet överst i labben
https://canvas.kth.se/courses/12205/assignments/99863?module_item_id=171679

Skriv ett program som simulerar kortleket som står beskrivet överst i tabben.

[Titta även på denna video](https://youtu.be/Njk4PUEICBQ) [_ \(https://youtu.be/Njk4PUEICBQ\)](https://youtu.be/Njk4PUEICBQ)



[\(https://youtu.be/Njk4PUEICBQ\)](https://youtu.be/Njk4PUEICBQ)

för att förstå hur programmet ska fungera.

Inmatningstips är att använda [input\(\)](https://docs.python.org/3/library/functions.html#input) [_ \(https://docs.python.org/3/library/functions.html#input\)](https://docs.python.org/3/library/functions.html#input) för att läsa in hela raden och [split\(\)](https://docs.python.org/3/library/stdtypes.html#str.split) [_ \(https://docs.python.org/3/library/stdtypes.html#str.split\)](https://docs.python.org/3/library/stdtypes.html#str.split) för att dela upp raden och [int\(\)](https://docs.python.org/3/library/stdtypes.html#int) för att konvertera till heltal. Experimentera sedan med olika inmatade ordningar och se om du kan lista ut i vilken ordning korten ska ligga innan man börjar trolle för att man ska få ut alla tretton i rätt ordning

[_ \(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#skapa-en-arrayq-modul\)](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#skapa-en-arrayq-modul) Skapa en ArrayQ-modul

Gör nu så här: klipp ut klassen från ditt program och klistra in i en ny fil arrayQFile.py

Importera klassen till huvudprogrammet med raden

```
from arrayQFile import ArrayQ
```

Använd if-satsen

```
if __name__ == "__main__":
```

i ditt huvudprogram för att inte köra eventuell kod, t.ex. testkod i ArrayQ. Nu går det att använda klassen utan att den syns i programmet. Skriv trollkarlsprogrammet med hjälp av din ArrayQ-klass.

[_ \(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#linkedq---en-k%C3%B6-av-noder-l%C3%A4nkad-lista\)](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#linkedq---en-k%C3%B6-av-noder-l%C3%A4nkad-lista)

LinkedQ - en kö av noder (länkad lista)

Nu ska du istället implementera kön som en länkad lista. Då behövs två nya klasser: Node och LinkedQ. Skriv in bägge klasserna i samma fil: linkedQFile.py. Noderna i listan är objekt som vardera innehåller två publika attribut: ett värde (value) och en referens till nästa objekt (next). Själva LinkedQ-klassen ska ha två privata attribut: first som håller reda på den första noden i kön och last som pekar ut den sista. Använd samma gränssnitt som i uppgift 4.

som pekar ut den sista. Använd samma gränssnitt som i uppgift 1:

enqueue(x) stoppa in något sist
x = plocka ut det som står
dequeue() först
isEmpty() kolla om kön är tom

Det är extra knepigt att programmera enqueue(x) eftersom det blir två fall, beroende på om kön är tom eller inte. Rita upp bägge fallen (lådor med pilar) innan du skriver koden!

[_\(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#testa-linkedq\)_](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#testa-linkedq) Testa LinkedQ

Prova din kö med följande testfall.

- Skapa en kö, kolla att den är tom
- Skapa en kö, lägg till och ta bort ett element, kolla att kön är tom
- Skapa en kö, lägg till tre element
 - Kolla att kön inte är tom
 - Plocka ur de tre elementen ett och ett och kontrollera att det är rätt element som returneras.

[_\(https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#trollkarlsprogramme-med-linkedq\)_](https://gits-15.sys.kth.se/tilpro-assar/betalabbar-tilpro/blob/master/trollkarlsko/d1.org#trollkarlsprogramme-med-linkedq) Trollkarlsprogrammet med LinkedQ

Ändra import-satsen i trollkarlsprogrammet så att du importerar klassen LinkedQ istället för ArrayQ. Provkör. Fungerade det? Då har du lyckats implementera den abstrakta datastrukturen kö på två olika sätt.

När allt fungerar som det ska bör du ta en extra titt på koden. Är den kommenterad och begriplig?

Vid redovisningen ska du kunna

- **Berätta hur pythons array-metoder append, insert, remove och pop fungerar.**
- **Förklara varför attributen ska vara privata.**
- **Rita upp hur dina metoder fungerar för bägge implementationerna av kön.**
- **Fungerar de två implementationerna likadant? Förklara eventuella skillnader.**

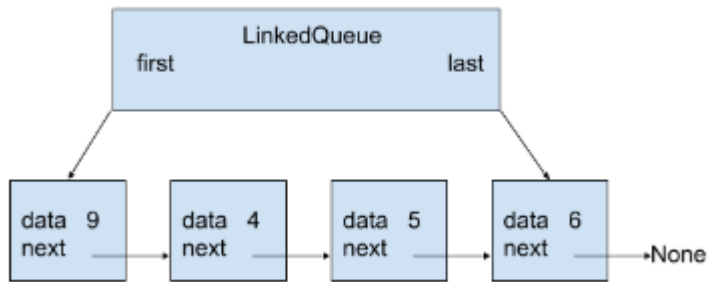
Att ta bort en specifik nod ur en kö

Skriv en metod i LinkedQ som tar bort en specifik nod ur en kö.

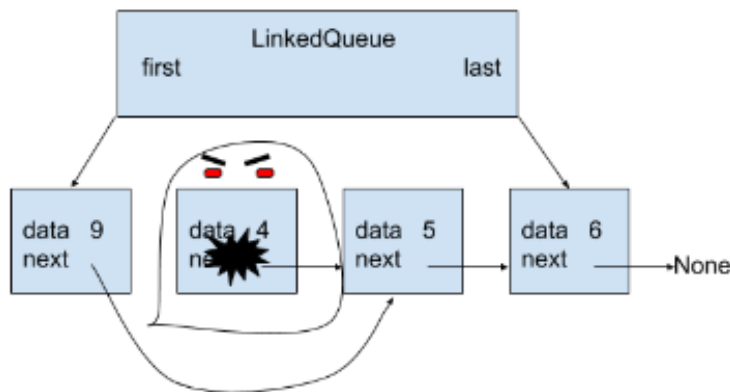
remove(x) tar bort x ur kön om x finns i kön

Om värdet (value) inte finns i kön så ska ingenting hända. Du tar bort noden genom att den

föregående nodens nextpekare sätts till efterföljande nod. Exempelvis om man ska ta bort noden med siffran 4 nedan.



Så måste, i normalfallet, nextpekaren i noden innan pekars om. Noden med siffran 4 blir en spökpöst som ingen pekar på och kommer så småningom att städas bort.



Om det är första noden som ska tas bort, så är det *first*-pekaren i LinkedQ som ska ändras. Testa din kod med följande fall

- Söka efter och ta bort det mittersta elementet i en kö med tre noder
- Ta bort första elementet i en kö med en nod
- Försök ta bort ett element som inte finns i en kö med några noder (kön ska förbli oförändrad)
- Försök ta bort ett element ur en tom kö (inget ska hända och programmet ska inte krascha)
- Ta bort sista elementet i en kö med några noder

Spara alla testfallen separat och visa dem vid redovisningen

Rita minnesbild

När man programmerar är det viktigt att förstå hur datorns minne ändras under programkörning. En minnesbild består i princip av namngivna lådor med värden i. På

<http://www.pythontutor.com/visualize.html>

(<http://www.pythontutor.com/visualize.html#mode=display>) kan du ladda upp pythonkod få minnet uppritat under körning. Rita av steg för steg hur minnet ser ut när man söker efter och tar bort det mittersta elementet i en kö. Visa din ritning under redovisningen.

Vid redovisningen ska du kunna

- Visa dina testfall för borttagning av element
- Visa minnesbild för borttagning av element.