class Queue:

    # max\_size: size of Q

    # Q: Array

    def \_\_init\_\_(self, max\_size):

        self.max\_size = max\_size

        self.Q = [0] \* max\_size

        self.num = 0

        self.first = 0

    def enqueue(self, item):

        if self.num >= self.max\_size:

            raise Exception("Queue overflow")

        self.Q[(self.num + self.first) % self.max\_size] = item

        self.num += 1

    def dequeue(self):

        if self.num == 0:

            raise Exception("Queue empty")

        item = self.Q[self.first]

        self.first = (self.first + 1) % self.max\_size

        self.num -= 1

        return item

    def search(self, i):

        if self.is\_empty():

            raise Exception("Queue empty")

        while self.first != self.num:

            if i == self.Q[self.first]:

                print("found it")

                return True

            self.first += 1

        print("not found")

        return False

    def front(self):

        if self.num == 0:

            raise Exception("Queue empty")

        return self.Q[self.first]

    def is\_empty(self):

        return self.num == 0

    def size(self):

        return self.num

    def is\_full(self):

        return self.num >= self.max\_size