

GIT Department of Computer Engineering

CSE 222/505

Spring 2016

Homework 06

Onur SEZER – 121044074

- Ödev Maven projesi olarak oluşturulmuştur.
- Exception Handling gerekli yerlerde kullanılmıştır.
- JUnit testleri yapılmıştır.
- Methodlar Javadoc kullanılarak implement edilmiştir.
- Ödev HuffmanTree, BinarySearchTree, ArrayListPriorityQueue, LinkedListPriorityQueue ve UnsortedVectorPriorityQueue klaslarından oluşmaktadır.
- **HuffmanTree classı:**
 - HuffmanTree clasında verilen source kodun üstüne encode methodu yazılmıştır.
 - Encode methodunda verilen stringi binary şekilde şifreleyip çıkan sonucu return edilir.
 - Encode methodunda parametre olarak gelen stringin karakterlerini tek tek wrapper fonksiyonuna gönderir.
 - Wrapper fonksiyonda recursive olarak binary tree yi dolaşır.
 - Binary treenin soluna giderse 0, sağına giderse stringe 1 eklenir.
 - Eğer leaf e gelmişse recursive den çıkar.
 - Leaf e geldiğinde leaf deki sembolle gönderilen karakter eşitse 1-0 lardan oluşan stringi gönderir ve çıkar. Değise null gönderir.
- **BinarySearchTree classı:**
 - BinarySearchTree classının iterator ü yazılmıştır.
 - next() ve hasNext() methoslari implement edilmiştir.
 - Iterator implement edilmeden önce, tree inorder yöntemiyle dolaşılıp sıralı elamanlar bir ArrayListe atılmıştır.
 - next ile eleman verilirken ArrayListteki sirali elamanlardan verilir.

- **ArrayListPriorityQueue classı:**
 - ArrayListPriorityQueue classı PriorityQueue interface ini implement eder.
 - Interface de enqueue, dequeue, size ve isEmpty methodlari bulunur.
 - Bu methodlari ArrayList ile implement eder.
 - Bu class implement edilirken JAVA nin PriorityQueue classindan faydalanilmiştir.
- **LinkedListPriorityQueue classı:**
 - LinkedListPriorityQueue classı PriorityQueue interface ini implement eder.
 - Interface de enqueue, dequeue, size ve isEmpty methodlari bulunur.
 - Bu methodlari ArrayList ile implement eder.
 - Bu class implement edilirken JAVA nin PriorityQueue classindan faydalanilmiştir.
- **UnsortedVectorPriorityQueue classı:**
 - UnsortedVectorPriorityQueue classı PriorityQueue interface ini implement eder.
 - Interface de enqueue, dequeue, size ve isEmpty methodlari bulunur.
 - Bu methodlari ArrayList ile implement eder.
 - Bu class implement edilirken JAVA nin PriorityQueue classindan faydalanilmiştir.

MAIN deki TEST:

=====

PART1

Encoded Codes :

c: 00000

u: 00001

h: 0001

r: 0010

s: 0011

e: 010

i: 0110

n: 0111

b: 100000
g: 100001
p: 100010
y: 100011
o: 1001
a: 1010
l: 10110
d: 10111
v: 1100000
j: 1100001000
q: 1100001001
x: 1100001010
z: 1100001011
k: 11000011
w: 110001
m: 110010
f: 110011
t: 1101
_: 111

Code to Message :

11000010011111110010100001 : q__rg

String => q__rg



“q__rg” sitringi parametre olarak
gönderilir.

Encode : 11000010011111110010100001



Encode methodundan gelen string

=====

PART2

2

1

0

null

null

null

12

11

null

null

20

null

null

2

1

0

null

null

null

12

11

null

null

20

15

null

null

null

2

1

0

null

null

null

12

null

20

15

null

null

null



sırasıyla ilk önce 2, 1 , 12, 20, 11, 0 eklenir. Next methodu ile tek tek elamanlari çekilir.

next : 0

next : 1

next : 2

next : 12

next : 15

next : 20

hasNext :false



Elemanların hepsi tek tek çekildiğinden hasNext false return eder.

=====

PART3

#ArrayList

isEmpty : true

size : 4



Dört tane eleman girilir, priority queuedeki durumları

3

4

10

100

isEmpty : false

dequeue yapıldı



dequeue yapılırken en üstteki elemanı siler

4

100

10

#LinkedList

isEmpty : true

size : 5



Beş tane eleman girilir, priority queuedeki durumları

1

2

10

8

20

isEmpty : false

dequeue yapildi



iki defa dequeue yapılır

dequeue yapildi

8

20

10

#UnsortedVector

isEmpty : true

size : 5



Beş tane elaman girilir, priority queuedeki durumları

4

8

5

20

30

isEmpty : false

dequeue yapildi

dequeue yapildi

dequeue yapildi



üç defa dequeue yapılır ve son durum aşağıdaki gibi olur.

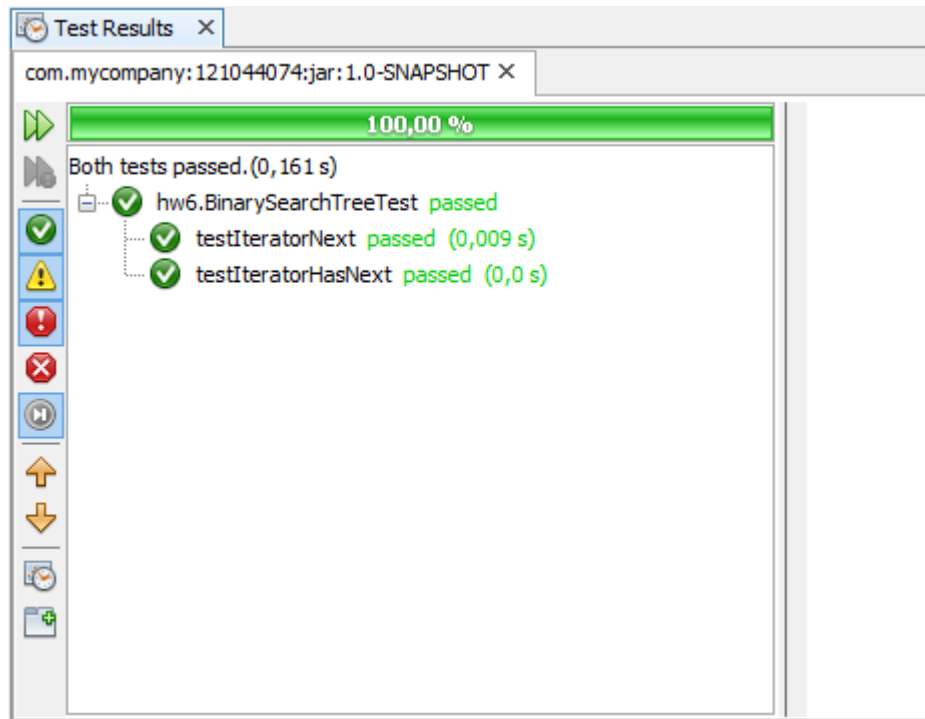
5

30

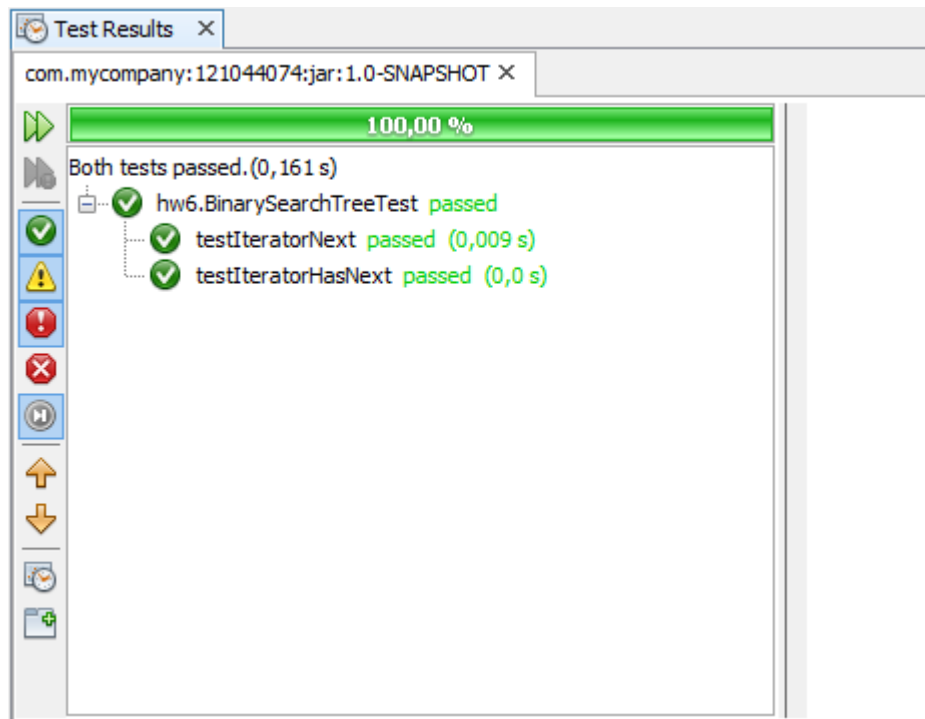
Process finished with exit code 0

JUnit TESTs

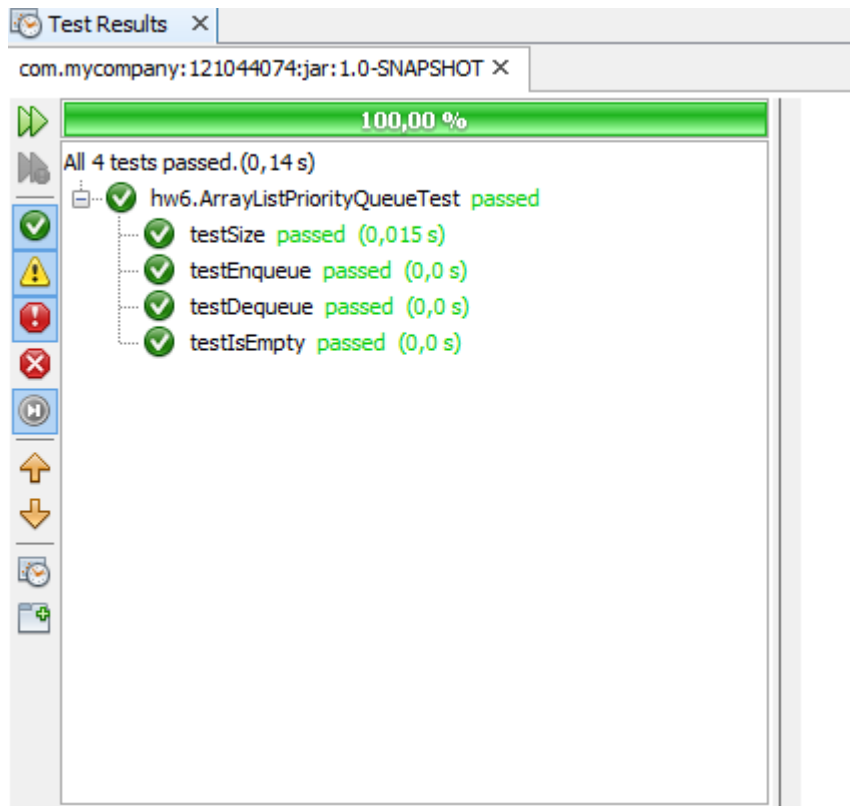
1. HuffmanTree



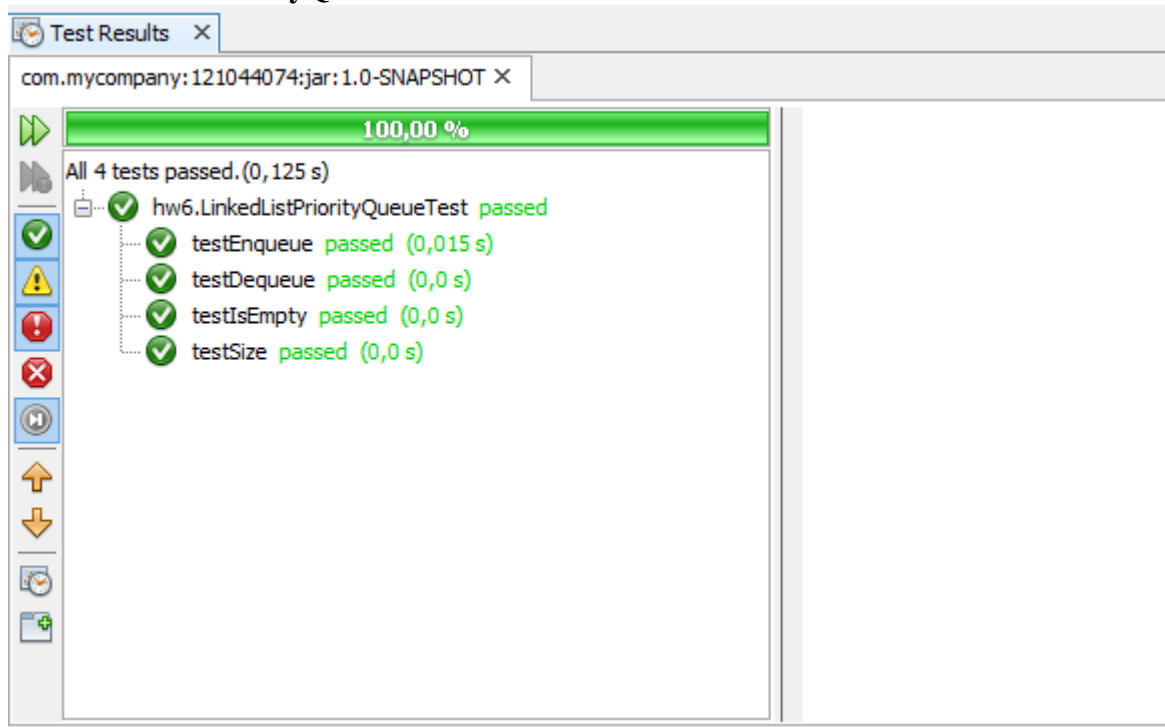
2. BinarySearchTree



3. ArrayListPriorityQueue



4. LinkedListPriorityQueue



5. UnsortedVectorPriorityQueue

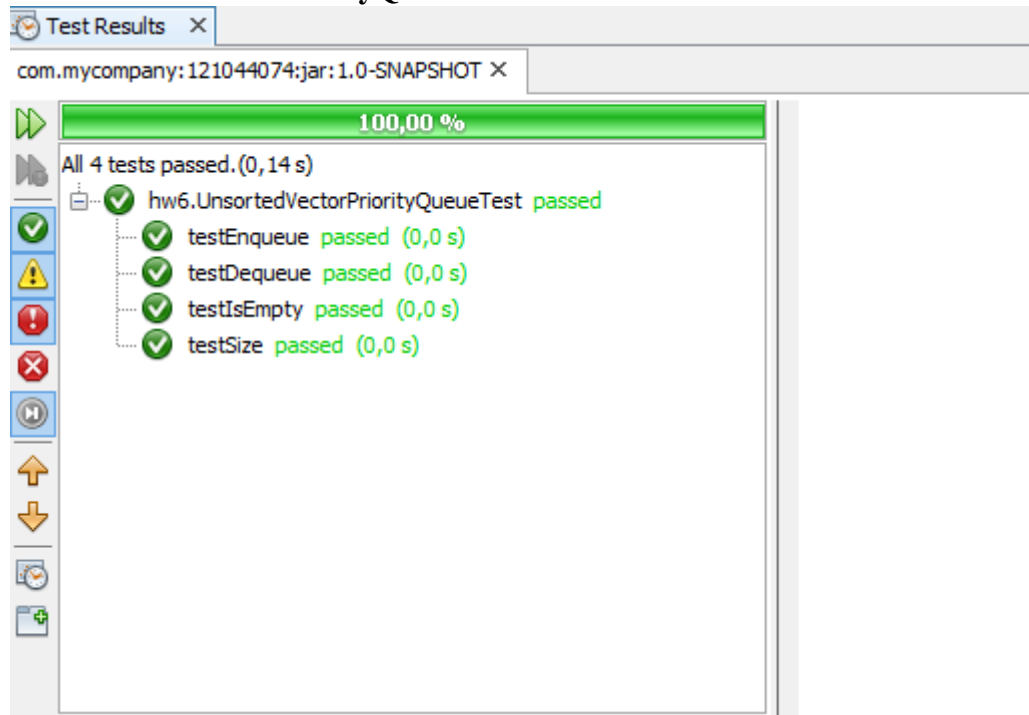


Diagram:

