Lecture23
Friday, 22 May 2020 7:07 PM

Generics - Stacks, Queues, LL, BT, BST } → limitation.

Stack, Queues, LL ⇒ Integer

A, B, C, D.

LL {
Node {
int data;
String data;
}
}

LL {
Node {
Student data.
}
Student {
name:
age.
}
}

I am implementing a new class each and every time

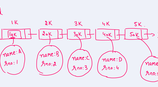
Resolution - Generics ⇒ General

⇒ linkedlist <T> {
Node {
T data;
Node next;
}
}

Problem { linkedlist<Integer> ll = new linkedlist<>();
... <String> ll; ___ ___
linkedlist <Student> ll; ___ ___

}

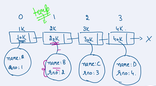
head



public void display() {
Node temp = this.head;
while(temp != null) {
System.(temp.data + " ⇒ ");
temp = temp.next;
}
}
Student {
String toString() {
return { this.name + ": " + this.rollno ";
}
}

10K

{A : 1} ⇒ {B : 2} ⇒ {C : 3}.



linkedlist <T>
↳ Student

Student tf = new Student("B", 2);

public int find (T data) {
int index = 0;
for (Node temp = this.head; temp != null; temp = temp.next) {
if (temp.data.equals(data)) {
return index
}
index++; // 1
}
return -1;
}

B

10K = 50K
20K = 50K.

Student {
public boolean equals(Object 50K other) {
Student s = (Student) (other);
return this.name.equals(other.name); //true
}
(B) . (B)
}

hashmap
↳ key
↳ value }

HashMap <String, Integer> map = new HashMap<>();

key value

String	Integer
India	300
Russia	200

map.put("India", 300);
⇒ map.put("Russia", 200);
⇒ map.put("USA", 150);

map.put("India", 300);
map.get("Russia"); //200
map.get("India"); //300
map.get("NZ"); ⇒ null.

keys are unique.

map.remove("USA"); // 150.
map.remove("NZ"); // null

map.containsKey("India");
↳ true
↳ false

put
get
remove
containsKey } ↳ O(1) time

↳ Solve many problems in O(n) time ↳

Q↳ Character with Max freq in a String?

String str = "aabccaadef";

↳ a → 3

↳ O(length of String);

Character	Integer
a	3
b	2
c	1
d	1
e	1
f	1

⇒

Entryset ⇒ maxChar → entry.getKey();

one → { 3, 1, 5, 6 } m
two → { 1, 5, 4, 7 } n
(m+n)

intersection
{ 1, 5 }

O(m+n)

nlogn + nlogn
(m+n) logn

Integer	Boolean
3	false
1	false
5	false
6	false

⇒ O(n).

Q↳ Subarray with Sum zero? ↳ Resume 9.00pm

arr = { 2, 5, 1, 8, -8, -1 }

↳ true
↳ false

↳ O(n) time

Subarrays

arr = { 3, -5, -8, 9, -1 }

① Generate all Subarrays ↳ O(n²)

② Prefsum approach.

0	1	2	3	4
3,	-5,	-8,	9	-1.

prefsum: 3, -2, -10, -1, -2

true

arr₁, arr₂, arr₃, arr₄ . . . . , arrₙ.

p₁, p₂, p₃, p₄, , pₙ

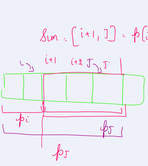
3, -5, -8, 9, , -1.

sum[0, 1] = p₂ = -2

sum[0, 3] = p₄

p₃ - p₁
-1 - (-2) = -1+2
= 1

sum = [i+1, j] = p[j] - p[i]



p₄ - p₂ = sum[i+1, j].

arr[i+1, j] = Subarray

p₄ - p₂ = 0

p₄ = p₂

Generics
Hashmaps
Questions on hashmaps

Tomorrow no class ⇒