

Exame P&R

①

I) Análise de complexidade

exemplo $\text{for (int } j=1; j < N; j+=2) \log_2 N$

exemplo (dependência de variáveis encadeadas)

$\text{for (int } i=0; i < N; i++)$

$\text{for (int } j=0; j < i; j++)$

$\text{for (int } k=0; k < j; k++)$.

} encadeamento
 $O(N^3)$

ma análise puxa de o maior

ciclo do while são ciclos de faz

exemplo

$\text{for (long } i=0; i < N; i+=2) \text{ tempo de execução } N/2$
 complexidade $O(N)$

$2N = N \Rightarrow$ análise de complexidade é $O(N)$, dei é uma constante

exemplo

$\text{for (long } j=0; j < N \times N; j++)$

$N^2 \Rightarrow$ complexidade $O(N^2)$

exemplo $\text{for (int } x=1; x < N; x*=2) \log N$

$\text{for (int } i=0; i < 5000; i++) \text{ constante}$

$\text{for (int } k=0; k < N; k++) N$

análise complexidade $O(N \times \log N)$

body move

the human body move

it will be the main thing in the world

it can do many things in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

(the human body move)

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

but it's not the best thing in the world

12) contrução de métodos simples

Geome Deco

jogar em do const

fase em do pesquisa ema ou varz vez

desenvolhe logaritmico / não existem loops / sem whiles

desenvolhe $O(N + \log M)$ // errado em while ??

十一月三十日 晴

Exemplo...

Exemplo Declar

parametro pode ser comparado com qualquer valor da lista

(5)

< T>xcid remocao (lit & Ts lista, comparable &? super T> valor)

o melhor desempenho: percorrer, tempo

Exemplo...

recebe deu valor e uma lista, ou valor padrao da comparação com a lista

< T>xcid orden (lit & Ts lista, <comparable &? super T> A

, <comparable &? super T> B) 2

Iterador < T> it = lista.iterator();

lit < T> temp = menor(lit & Ts lista);

T xl;

while (it != null.next()) {

xl = next();

if (xl.comparavel (B) >= 0) { temp.add(xl); }

if (xl.comparavel (A) < 0 || xl.comparavel (B) > 0) temp.add(xl); }

list.clear();

list.addall(temp);

}

Exercice Devoir

(6)

exercice ...

mettre les nœuds d'un arbre dans une liste

en valeur (on va faire un comparaison entre la valeur du nœud

et T) et puis (comparable < ? seign TSS void) ((T > T) liste,

T min ; T max)

Il faut : $IT \leftarrow IT \cup \{ \text{list}.it.push() \}$;

$(IT < T) \text{ temp} \leftarrow \text{new linkedlist}();$

$T \leftarrow \text{temp};$

while (IT. itomxt() / {

temp = itomxt();

if (min. comparable (temp) >= 0 && max. comparable (temp) <= 0) {
 temp. add (temp);

}

list. clear();

list. addall (temp);

}

U. class Iterator
(contains Iterator)

Exercise Deco

example... class iterated on may have name & mark
not a possible action value

contains Iterator (with separate function or exception)
private class Person implements Iterable<String> {

String name, mark;

public Person (String n, String p) {

this.name = n;

}
@override

public Iterator<String> iterator() {

return new PersonIterator(this);

private class PersonIterator implements Iterator<String> {

int pos = 0;

Person person;

public PersonIterator (Person p) {

this.person = p;

}
@override

@override

public boolean hasNext() {

public String next() {

if (!hasNext()) { throw new NoSuchElementException(); }

if (pos < 0 = -1) {

pos++

return person.name;

if (pos = 2) {

pos++

return person.mark;

return null; } }

exemplo...

Exercício 8

classe para armazenar em memória
não é possível alterar o valor
contém o iterator

public class Intervor implements Iterable<Inteiro> {

int numero;

public Iterator<Inteiro> iterator() {

 this.numero = n;

}
@override

public Iterator<Inteiro> iterator() {

 return new IteratorInteiro(this);

 public int get() {

 return numero;

}
}

public class IteratorInteiro implements Iterator<Inteiro> {

 Inteiro inteiro;

 int posicao = 0;

 public IteratorInteiro(Inteiro x) {

 this.inteiro = x;

}

@Override

 public boolean hasNext() {

 return posicao < 1;

}

@Override

 public int next() {

 if (!hasNext()) throw new NoSuchElementException();

 posicao++

 return Inteiro.get();

}

@Override

 public void remove() { throw new UnsupportedOperationException(); }

a) classe que fazem implementos `IComparable`

com construtor

com set

com @override public `IComparable<tipos>` compareTo(),
return new `IComparable<tipos>`;

}

• no iterator

@override

return `getNext()`

@override do

type `next()`

if (`!hasNext()`) throw new NoSuchElementException();

@Override

void remove()

throw new IllegalStateException()

return sua @remove

}

exemplo: alternativa com que os métodos de iterator fazem para o
classe principal

class com implements `IComparable<...>`

public `IComparable<...>` iterator()

`IComparable<...>` it = new Iterator{

@override

return sua `getNext()`

}

@Override

public int next() { ... }

@Override

int compare (int l1, int l2) { ... }

@Override

void add(...) { ... }

@Override

void remove(...) { ... }

Matthews (no)

+31,000

Wetland area

Abstract 20

2011-07-08

1000

ANSWER: Cetaceans are mammals that live in water.

1990

41 years old

Chlorophyll in our work

NaCl - Osmotic pressure

大山地区に特有の温泉

• What is the relationship between the two types of energy?

8. *Leucosia* (Leucosia) *leucostoma* (Fabricius)

May 20

8. 100

2020-07-09

1. 100% 2. 100% 3. 100% 4. 100%

2 - 5 May 1965

2010-0

6.3 Mapas

(Excepciones)

HashMap Mapa:

Set (key) - recoge a Key devolviendo valor

put (key, value) - adiciona

key () - devuelve el conjunto Set con los claves de Map

remove (key) - remueve

containsKey (key) - pregunta

Ejemplo:

Centro de impresión:

```
public class GestorDeImpresion {
```

```
    Hashmap MP = new Hashmap < >();
```

```
    public Impresion getImpresion (String name) {
```

```
        return MP.get (name);
```

```
}
```

Obtener todos

```
    return MP.keySet()
```

agregarlos

```
    MP.put (i.getName (), i);
```

remover

```
    MP.remove (name)
```

Indica si existe

```
    return MP.containsKey (name)
```

Wiederholung

Wiederholung

zurück, nach unten

zurück, nach unten - (rot) top

zurück - (rot) top

zurück, zurück, zurück, zurück, zurück - (rot)

zurück - (rot) zurück

zurück - (rot) zurück

zurück, zurück, zurück - (rot)

zurück, zurück, zurück - (rot)

zurück, zurück, zurück - (rot)

zurück, zurück, zurück, zurück, zurück - (rot)

zurück, zurück, zurück, zurück, zurück - (rot)

(wiederholung, wiederholung)

[2] metoda priorityQueue / fila de espera

Exercícios

(12)

metodo disponivel

add (object) // adiciona trabalho

poll // remove trabalho

peek - consulta o ultimo elemento

remove (key)

isEmpty // se a fila tem alguma cosa

size // numero de trabalhos

exemplo (adiciona um elemento ordenado)

PriorityQueue <Trabalho> Pn = new PriorityQueue<>(new ComparadorTrabalhos());

public int compare (Trabalho a, Trabalho b) {

return a.getNroPag() - b.getNroPag();

} ;

Wetland

soil & air movement reduce

8)

living in wet soil

allow more water infiltration

and prevent flooding

allow water to soak in faster

(a) water

water will soak in faster

allow water to soak in faster

water soak in faster

soil absorb water & prevent flooding

allow water to soak in faster

absorb water & prevent flooding