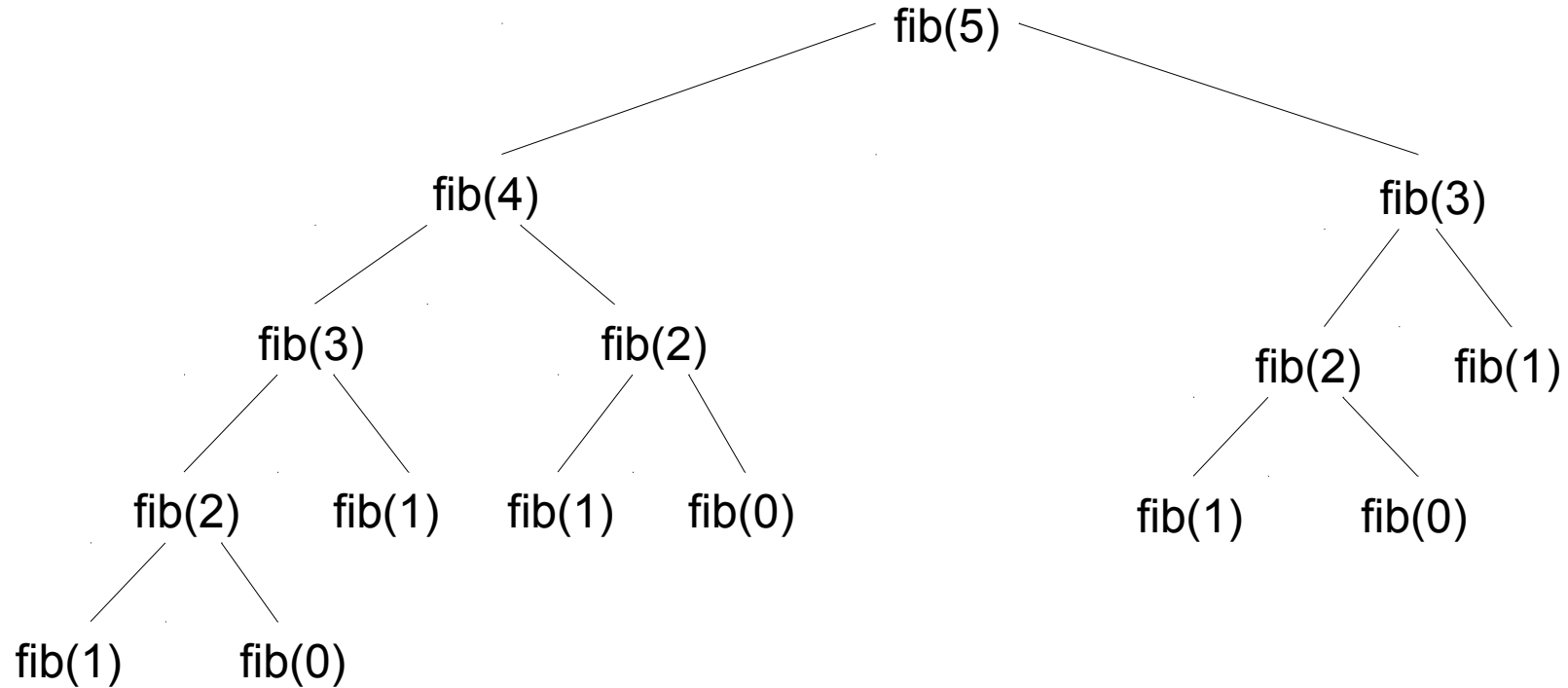


Tree of recursive calls for fib(5)



$$1 + 2 + 4 + 8 + 16 + \dots$$

$$= 2^0 + 2^1 + 2^2 + \dots + 2^{n-1}$$

$$= 2^n - 1$$

$$= O(2^n)$$

$$\text{fib}(200)$$

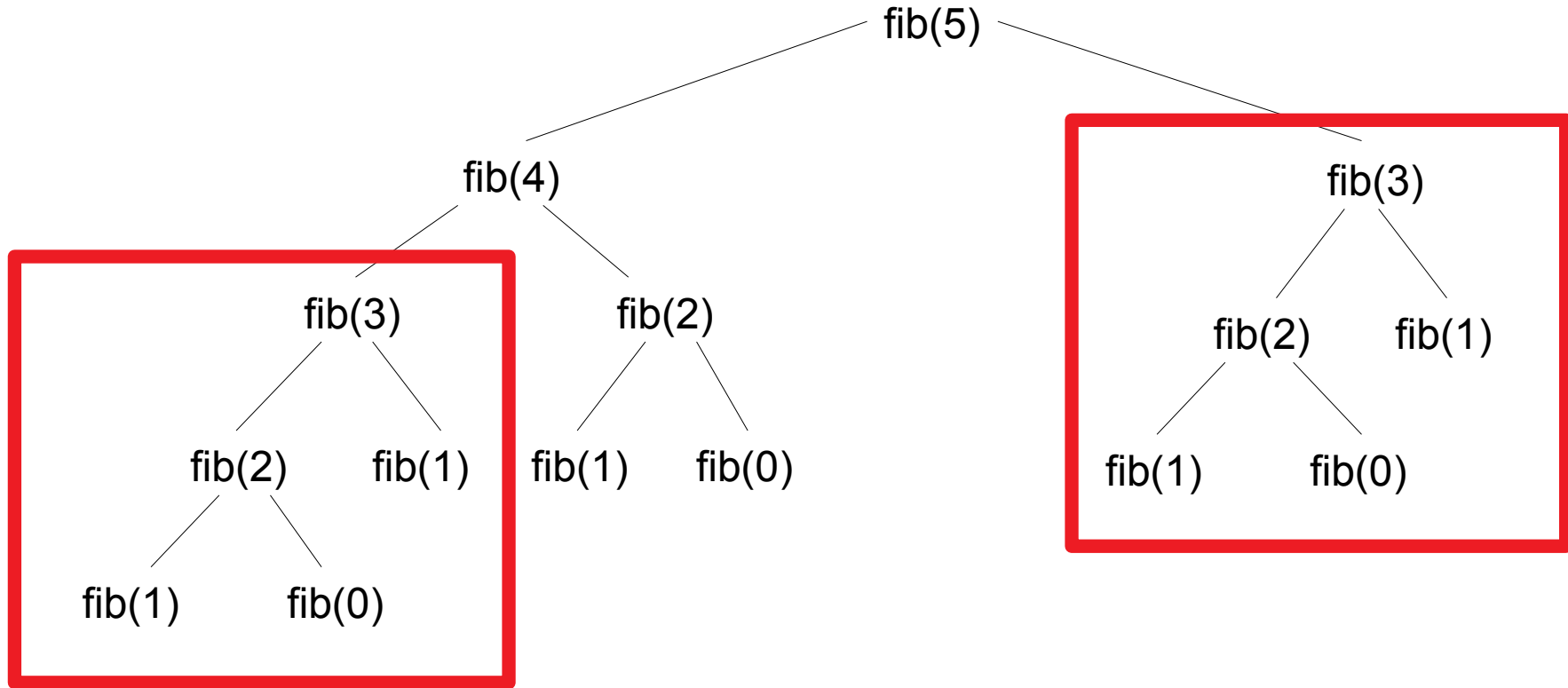
$$= 2^{138} \text{ recursive calls}$$

$$= 2^{92} \text{ seconds}^*$$

$$= 1.56915162 \times 10^{20} \text{ years}$$

* assuming 40 trillion operations per second

Tree of recursive calls for fib(5)



HashMap<Key, Value>

Fibonacci:

Key: input n (Integer)

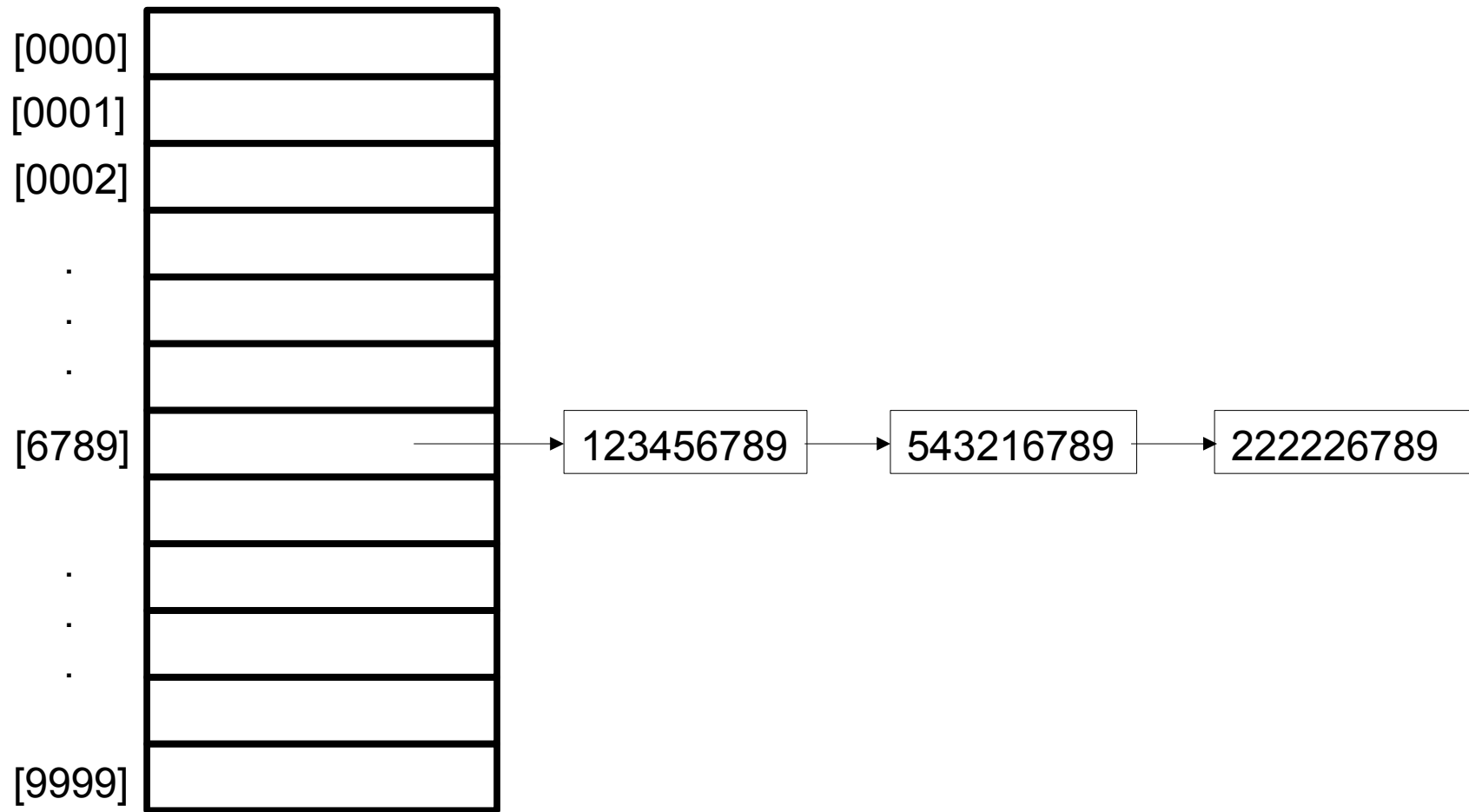
Value: answer (Long)

WPI

Key: SSec (Integer)

Value: Student object

```
class Student {  
    String name;  
    int ssec;  
    double gpa;  
}
```



	Open addressing with linear probing	Open addressing with double hashing	Chained hashing
Load factor (α)	$\frac{1}{2} \left(1 + \frac{1}{1 - \alpha} \right)$	$\frac{-\ln(1 - \alpha)}{\alpha}$	$1 + \frac{\alpha}{2}$
0.5	1.50	1.39	1.25
0.6	1.75	1.53	1.30
0.7	2.17	1.72	1.35
0.8	3.00	2.01	1.40
0.9	5.50	2.56	1.45
1.0	Not applicable		1.50
2.0			2.00
4.0			3.00

User-Defined Keys

- If your key is a user-defined type, you must define `hashCode()` and `equals()` on the key.
- If `item1.equals(item2)` then it must be true that `item1.hashCode() == item2.hashCode()`
- If `item1.hashCode() == item2.hashCode()`, it's not necessarily true that `item1.equals(item2)`