Com S 228

Spring 2014

Exam 1 Sample Solution

1.

`BigCat mufasa = new Lion("Mufasa", Sex.MALE);` `mufasa.speak();`	`Roar!`
`Interspecies das = new Tiger("Das", Sex.MALE);`	`compile error: Cannot convert from Tiger to Interspecies`
`IRoar kofi = new IRoar();` `BigCat thema = new BigCat("Thema", Sex.FEMALE);`	`compile error: Cannot instantiate the type IRoar or BigCat`

`BigCat sanjeev;`
`sanjeev = new Tiger("Sanjeev", Sex.MALE);`
`sanjeev.speak();`
`sanjeev = new Liger("Vijay", Sex.MALE,`
`  new Lion("Simba", Sex.MALE),`
`  new Tiger("Maha", Sex.FEMALE));`
`sanjeev.speak();`
	`Growl!`
	`Roar-Growl!`

`IRoar nala = new Lion("Nala", Sex.FEMALE);` `Tiger rita = (Tiger) nala;`	`ClassCastException`

`BigCat vijay, nala;`
`vijay = new Tiger("Vijay", Sex.MALE);`
`nala = new Lion("Nala", Sex.FEMALE);`
`Interspecies ife = new Tigon("Ife", Sex.MALE,`
`  (Tiger) vijay, (Lion) nala);`
`ife.getParents();`
	`Dad: Tiger (Vijay)`
	`Mom: Lion (Nala)`

`IRoar nala;`
`nala = new Lion ("Nala", Sex.FEMALE);`
`nala.getParents();`
	`compile error: getParents() undefined for IRoar`

2a)

```
public boolean equals(Object another)
 {
       // check if another is null
       if (another == null) return false;

       // check the type of another
       if (!(another instanceof Dictionary)) return false;

       Dictionary d = (Dictionary) another;

       // check null word
       if (word == null && d.word == null) return true;

       if ( (word == null && d.word != null) ||
              (word != null && d.word == null) )
              return false;

       // neither name nor and another.getName() is null
       // check length
       if (word.length != d.word.length) return false;

       // check words one by one
       for (int i=0; i<word.length; i++)
       {
              if (!equals(word[i], d.word[i]))
              {
                     return false;
              }
       }

       return true;
 }
```

b)

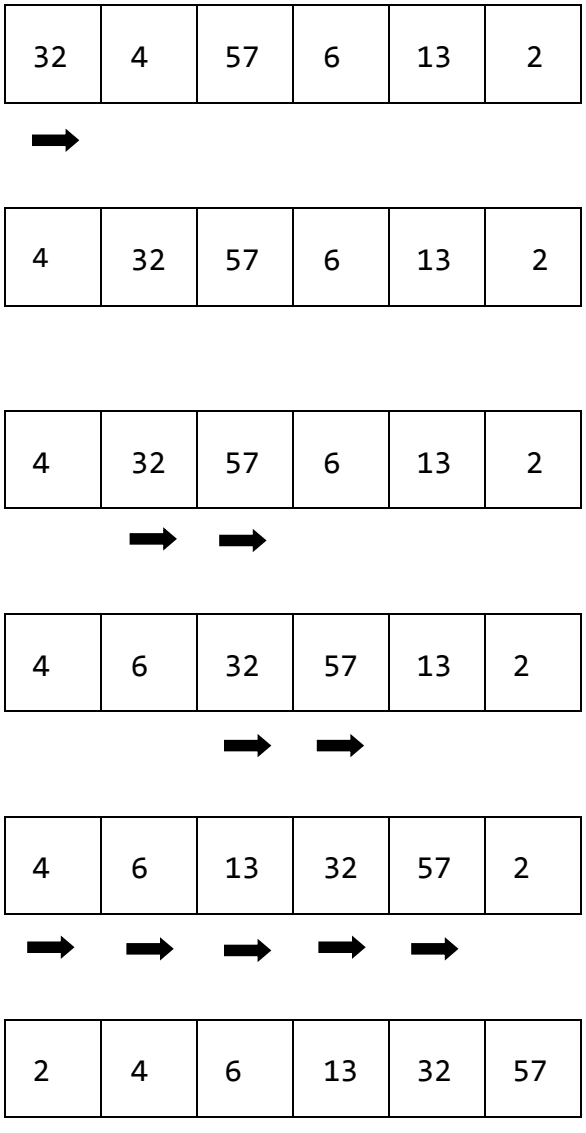
```
public Dictionary makeClone()
{
      // TODO
      if (word == null)
      {
            return new Dictionary(null);
      }

      String[] wordCopy = new String[word.length];
      for (int i=0; i<word.length; i++)
      {
            wordCopy[i] = new String(word[i]);
      }

      return new Dictionary(wordCopy);
}
```

3a) $O(n)$ (exactly n); $O(n)$ (exactly $n - i$); $O(n^2)$
b) $O(n)$ (exactly $n - 1$); $O(n)$
c) $O(\log n)$; $O(n^2)$; $O(n^2 \log n)$
d) $O(n \log n)$

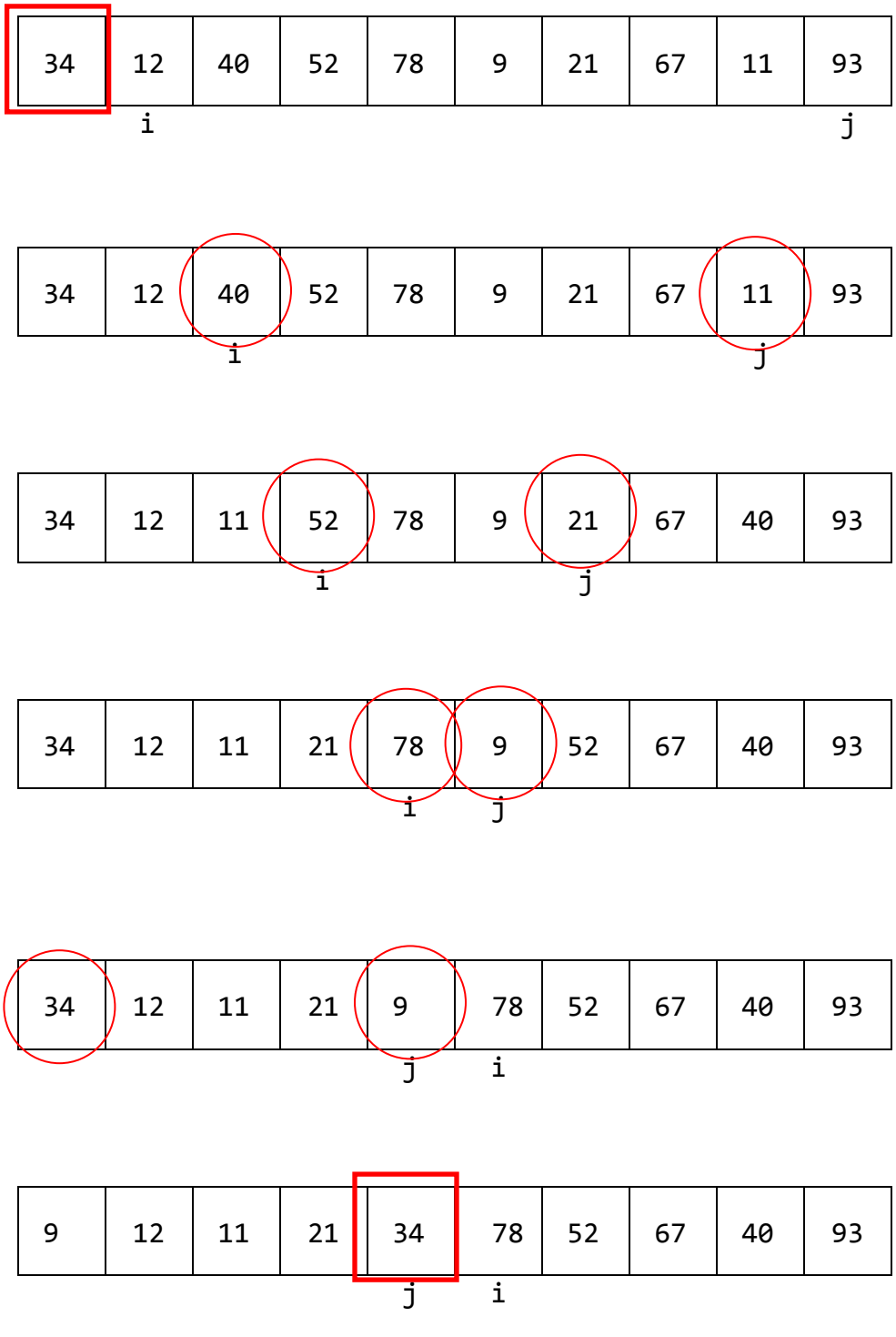
4a)



b) 10 right shifts (shown by the arrows above)

c)

```
pivotIndex == 2
```



d) A worst case happens when the n elements are already sorted, with running time $O(n^2)$.