ADSSE Re-exam 2022 Answers

1. a) True
   b) False
   c) False
   d) False
   e) True
   f) True
   g) False
   h) True
   i) True
   j) False

[1 mark for each correct part]

2. a) $T(n) = \Theta\left(n^{\frac{1}{2}} \lg n\right)$
   b) $T(n) = \Theta(n^3)$
   c) $T(n) = \Theta(n)$
   d) $T(n) = \Theta(n^2 \lg n)$
   e) $T(n) = \Theta(n^2)$

[2 marks for each part]

3.

a) Comparison sort requires $\Omega(n \lg n)$ operations in the worst case.

b) Examples include heap sort, merge sort and introsort.

c) i. Array B. ii. C[i] is the number of elements in the input less than or equal to i.

d)

```
RADIX-SORT(A, d)
for i = 1 to d
    use a stable sort to sort array A on digit i
```

e) Because otherwise, the sorting achieved on each iteration could be lost on the next iteration.

[2 marks for each correct part]

4.

a) O(n)

b) O(1)

c) O(1)

d) O(n)

e) O(1)

[2 marks for each correct part]

5.

a) \Theta(lg n)

b) 2 5 5 6 7 8 (or “nothing, because there is no return statement” – we have to allow this alternative answer because the question asks what the algorithm “returns” not what it “prints out”)

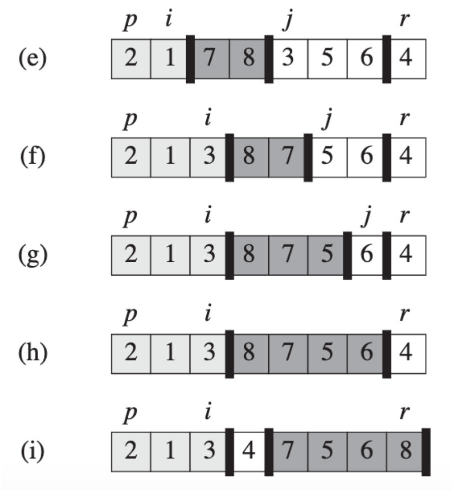
c) $\Theta(n)$

d) $\Theta(n)$

e) $\Theta(\lg n)$

[2 marks for each correct part]

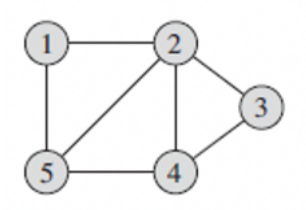
6.



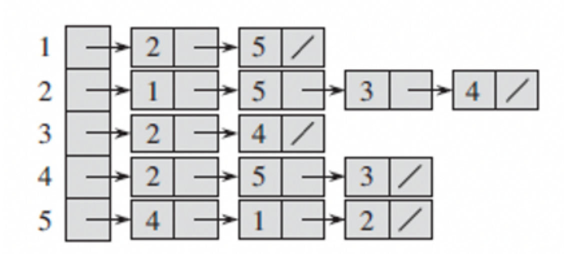
[10 marks, 2 for each iteration]

7.

a)



b)



c) O(E+V)

d) O(V^2)

e) Adjacency list, because if E is low, then E+V is much less than V^2.

[2 marks for each correct part]

8.

a)

```
STACK-EMPTY(S)
    if S.top == 0
        return TRUE
    else return FALSE
```

or just
Stack-Empty(S)
    return S.top == 0

[3 marks]

b)

```
PUSH(S, x)
    S.top = S.top + 1
    S[S.top] = x
```

[3 marks]

c)

```
POP(S)
    if STACK-EMPTY(S)
        error "underflow"
    else S.top = S.top - 1
        return S[S.top + 1]
```

[4 marks]

9.

a) If the keys are independently and uniformly distributed across the interval [0,1).

b) $h(k) = k \bmod m$

c) Because if $m = 2^p$ then h(k) is just the *p* lowest-order bits of *k*. Unless we know that all low-order *p*-bit patterns of *k* are equally likely, then it is better to make the hash function depend on all the bits of *k.*

d) A prime number not too close to a power of 2 but fairly close to n/α where α is the load factor and *n* is the number of elements to be stored.

e)

We can design a universal class of hash functions as follows:

1. Choose a prime number p greater than the largest possible value of k. Let $\mathbb{Z}_p = \{0, \ldots p-1\}$ and $\mathbb{Z}_p^* = \{1, \ldots p-1\}$. We assume $p > m$, since p is greater than the maximum value of k
2. We then define the universal class of hash functions

$$\mathcal{H}_{pm} = \{h_{ab} : a \in \mathbb{Z}_p^* \text{ and } b \in \mathbb{Z}_p\}$$

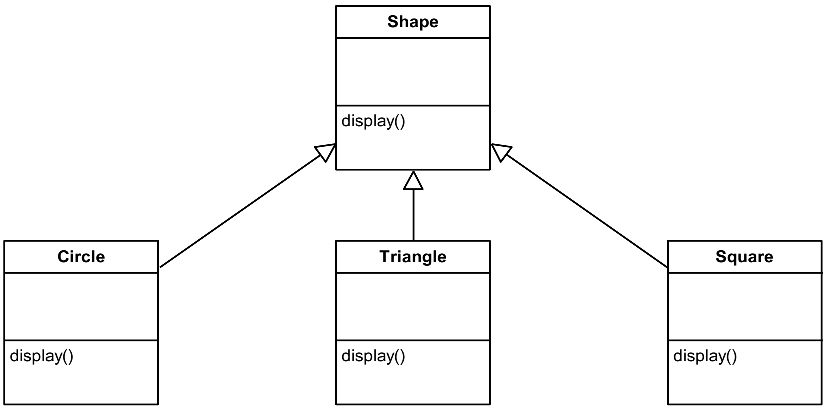
where

$$h_{ab}(k) = ((ak + b) \bmod p) \bmod m$$

[2 marks for each correct part]

10

a) Liskov substitution principle states that if *p* is a variable that refers to an object of class *x*, then *p* may refer to any object from any subclass of *x*.



For example, in the above diagram, if *p* refers to an object of class Shape, then it can also refer to any object whose class is Circle, Triangle or Square.
[4 marks]

b) Observer pattern [2 marks]

c) CompositeView should be a subclass of View. Example of the Composite pattern.
[4 marks]