1. a)

I = 0

minPos = 0

j = 1

j=2

j=3

j=4

minPos = 4

temp = 6

a[4] = 22

a[0] = 6

I = 1

minPos = 1

j=2

minPos = 2

j=3

j=4

minPos = 4

temp = 22

a[4] = 91

a[1] = 22

I =2 and so on….

6, 91, 24 ,78, 22, 89, 98, 99

6,22,24,78,91,89,98,99

6,22,24,78,91,89,98,99

6,22,24,78,89,91,98,99

b) midPos = 4

b = int[4]

c = int[4]

i=0

b[0] = 22

i=1

b[1] = 91,

i=2

b[2] = 24

i=3

b[3] = 78

I = 4

C[0] = 6,

I = 5

C[1] = 89

I=6

C[2] = 98

I =7

C[3] = 99

And so on….

c)

middle = 3

pivot = 78

I =0

J = 7

I = 1

J =6

J=5

J=4

Temp = 91

A[1] = 6

A[2] = 91

And so on…

2a)

Time complexity is how long it takes for the algorithm on average to take with the increasing amount of data n

b) space complexity is how much space is needed for the algorithm on average to take up with the increasing amount of data n

c) n/2 = .5n

.5n/2 = .25n

.25n/2 = .125n

1/(2^(h-1)) \*n is the number of elements at the bottom in terms of height h

Because this is an exponential function, if we do some arranging it’s clear that it’s going down by a factor of 2 or going down exponentially. In this case, this would be a log(n)