Review Tutorial 2 – Problem 1

a) Hand trace the Binary Search algorithm while we are searching for the **target 4** in the following data list:

| -27 | -14 | -2 | 1 | 2 | 3 | 4 | 6 | 8 | 9 | 13 | 24 | 35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Let's show our work.

Iteration 1:

Middle

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -27 | -14 | -2 | 1 | 2 | 3 | 4 | 6 | 8 | 9 | 13 | 24 | 35 |

Target 4 = middle element 4? Yes! Found it! We return its position i.e., its index 6.

Which scenario does the above situation represent: best, average or worst case scenario? Best case scenario -> algorithm did minimum amount of work (executed the critical operations (iterating and comparing) a minimum amount of time -> 1.

b) Hand trace the Binary Search algorithm while we are searching for the **target 13** in the following data list:

| -27 | -14 | -2 | 1 | 2 | 3 | 4 | 6 | 8 | 9 | 13 | 24 | 35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Let's show our work.

Iteration 1:

Middle

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -27 | -14 | -2 | 1 | 2 | 3 | 4 | 6 | 8 | 9 | 13 | 24 | 35 |

Target 13 = middle element 4? No! And since 13 > 4, we shall discard all elements left of 4 as well as 4.

Iteration 2:

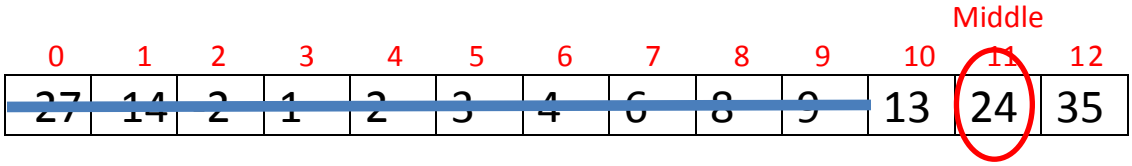
Middle

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ~~-27~~ | ~~-14~~ | ~~-2~~ | ~~1~~ | ~~2~~ | ~~3~~ | ~~4~~ | 6 | 8 | 9 | 13 | 24 | 35 |

The partition with elements 6 to 35 contains an even number of elements so I shall pick the element to the left of the mid section, i.e., 9 as the middle element.

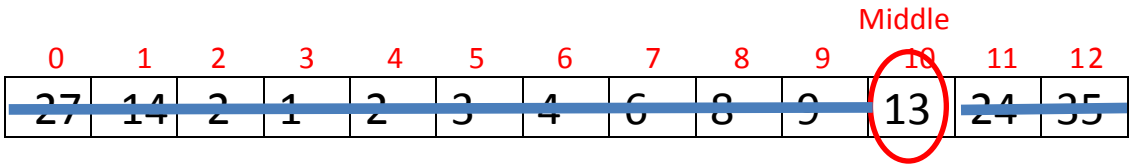
Target 13 = middle element 9? No! And since 13 > 9, we shall discard all elements left of 9 as well as 9.

Iteration 3:




Target 13 = middle element 24? No! And since 13 < 24, we shall discard all elements right of 24 as well as 24.

Iteration 4:




Target 13 = middle element 13? Yes! Found it! We return its position i.e., its index 10.

Which scenario does the above situation represent: best, average or worst case scenario? Worst case scenario -> algorithm could not partition anymore, i.e., the partition holding 13 contains only 1 element and cannot be partitioned further.
In other words, the algorithm did maximum amount of work, i.e., executed the critical operations (iterating and comparing) the maximum amount of time.

c) Hand trace the Binary Search algorithm while we are searching for the **target 5** in the following data list:

| -1 | -27 | 24 | -2 | 3 | 35 | -14 | 6 | 13 | 9 | 8 | 2 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Do you have any observations?
Yes, we cannot use the Binary Search on the above data list because its precondition (data list is sorted) is not satisfied. The result of the Binary Search will be unpredictable.