2-b)

```
Part B (int arr, arr_len) {
        int fillarr[];
        for(i=1; i<arr_len-1; i++) {
                if(arr[i] <= arr[i+1] && arr[i] <= arr[i-1])
                        fillarr.add(i);
        }
        return fillarr;
}
```

$T_{Best}^{(n)} = T_{Average}^{(n)} = O(n)$

$T_{worst}^{(n)} = O(n^2)$ In worst case dynamic fill array must be reallocated and reallocated $O(n)$ so total Algorithm complexity will be $O(n^2)$ in worst case.

3-)

```
int find(int arr, int arr_len, int target) {
        for(i=0; i<arr_len; i++) {
                for(j=1; j<arr_len; j++) {
                        if(arr[i] + arr[j] == target)
                                return 1;
                }
        }
        return -1
}
```

$T_{Best}^{(n)} = O(1)$ In best case arr[0] + arr[0] can be equal target, in this case loops only iterate 1 time so complexity will be $O(1)$

$T_{Average}^{(n)} = T_{worst}^{(n)} = O(arr\_len^2) = O(n^2)$