COMP9101 Ass02
Name: Zanning Wang
zID: z5224151

Q4:

In order to make the stack strictly increasing, We need to make sure that the first half is an arithmetic increasing sequence, so that we can get as many blocks as possible to fill in the insufficient ones later. In other words, if the total number can satisfy the Arithmetic increasing sequence, such as 1,2,3,4……,n-1,n; such movement must exist, the sudo-code may show like below:

```
1  let h = [h1, h2, h3,...,hi];
2  let total = 0;
3  let least = (0 + i-1) * i / 2;
4  for (let k=0; k<=i; k++) {
5    total += h[k];
6  }
7  if (total >= least) {
8    return true
9  }else {
10   return false
11 }
```

For line 3, we calculate the least number wo should have for those stack, which is (0 + i-1) * i / 2, the time complexity for this step is O(1);
For line 4 – 6, we calculate the total blocks we have, which the time complexity is O(n);
For line 7-10, check the count of total number if exceed the least requirement, which time complexity is O(1).
Therefore, the total time complexity is O(n);