hw5

資工三 110590002 王熿竑

1.

1.a. worse case time complexity: $\Theta(n^2)$

when all element place in the same bucket,
and the bucket is sorted by selection sort,
then the time complexity is $\Theta(n^2)$

1.b. simple change make worst case time complexity to $O(n\log(n))$

use quick sort to sort the bucket,
then the worst-case time complexity is $O(n\log(n))$

2.

2.a. algorithm

get length from 1 to n,
get max price in each length by split each length into two parts and get the max price in each part,
then get the max price in length n.

```
maxPriceInLength[n]
for i in 1 to len(rod):
  for j in 1 to i:
    maxPriceInLength[i] = max(maxPriceInLength[i], price[j] + maxPriceInLength[i-j])
return maxPriceInLength[n]
```

2.b. time complexity

$O(n^2)$

3.

input is length n
split the input into A[0,i] and A[i+1,n]
then for each char in A[0,i] find the mirror char in A[i+1,n] -> O(n^2)
i can at from 0 to n-1 -> O(n)

3.a. time complexity

$O(n) * O(n^2) = O(n^3)$

4.

4.a. algorithm

from last server S_n to first server S_1,
for all Server S_x
if have copy then cost is 0
else
calculate the cost of place a copy and
the cost of get a copy from S_j where have copy and closest,
then choose the minimum cost.

it can keep last server position to boost speed

4.b. time complexity

$O(n)$