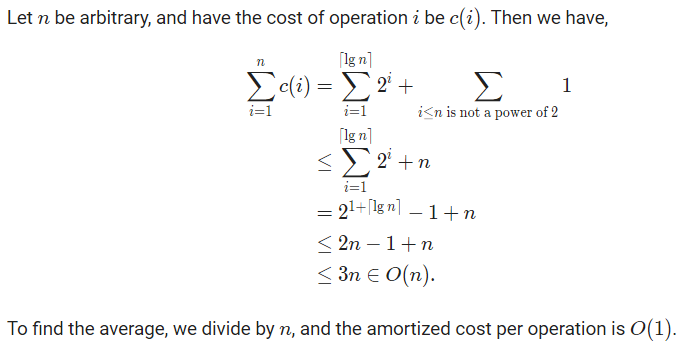
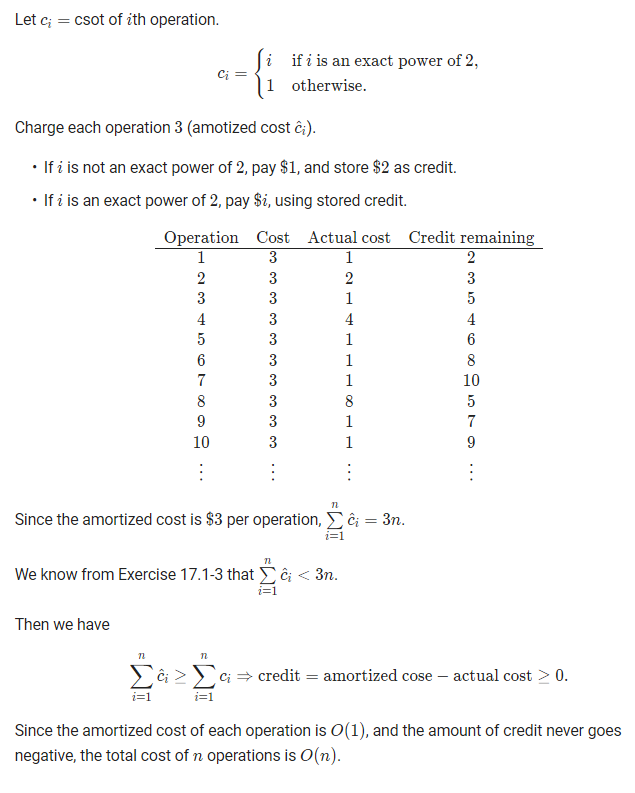
1、

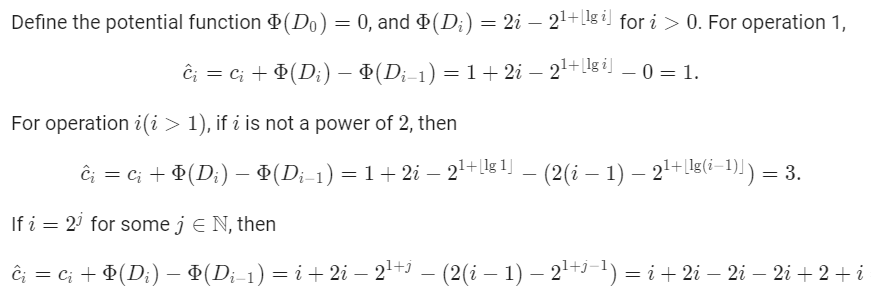
聚集法



会计法：



势能法：



So the amoritized cost is 3 per operation.

2、

聚集法

设第i个操作的代价为c(i)则：



因此平摊的代价是O(1)

会计法：

设第i个操作的代价为c(i),则：



每次操作花费3，若i不是2的幂，花费i+1，否则花费1存下2。

|  |  |  |  |
| --- | --- | --- | --- |
| 操作 | 支付 | 实际花费 | 存款 |
| 1 | 3 | 2 | 1 |
| 2 | 3 | 3 | 1 |
| 3 | 3 | 1 | 3 |
| 4 | 3 | 5 | 1 |
| 5 | 3 | 1 | 3 |
| 6 | 3 | 1 | 5 |
| 7 | 3 | 1 | 7 |
| 8 | 3 | 9 | 1 |
| 9 | 3 | 1 | 3 |
| 10 | 3 | 1 | 5 |

因为存款一直不为负数，所以：。所以平摊代价为O(1).

势能法：

Define a potention function Φ ; S.num is the number of objects in the stack , S.size is the size of stack.

Φ(S)=2\*S.num-S.size .

Let num**i** is the number of objects in the stack after the ith operation. Let size**i** is the size of stack after the ith operation.

If i is not the power of 2:

c^i=c**i**+Φ**i**-Φ**i-1**=1+(2\* num**i**-size**i**)-(2\* num**i-1**-size**i-1**)=1+(2\* num**i**-size**i**)- (2\* (num**i**-1)- size**i**)=3

If i is the power of 2:

c^i=c**i**+Φ**i**-Φ**i-1**=

num**i**+(2\* num**i**-size**i**)-(2\* num**i-1**-size**i-1**)= num**i**+(2\* num**i**-2\*(num**i**-1))- (2\* (num**i**-1)- (num**i**-1))=3

The amortized cost of Flipping\_push() is 3.