

2. ใน AI engine บางตัวไม่มีรูป ทุกสิ่งทุกอย่างในโลกนั้นต้องเขียนโปรแกรมด้วย recursive function สถานเดียว ในโลกนั้นมีเพียง Stack ให้ใช้โดยธรรมชาติ ไม่มี Queue ให้ใช้จึงเขียนโปรแกรมเพื่อสร้าง Queue ด้วย recursive function จากการ reuse ของ push หรือ pop operation พร้อมกับวิเคราะห์ว่า Queue ที่สร้างจาก Stack นั้น แพงกว่าเดิมเท่าไร สามารถตั้งสมมุติฐานได้เลยว่ามี Stack อยู่แล้ว แค่เขียน Enqueue กับ Dequeue ก็พอ

Time complexity Queue by stack ;

$$\text{enQueue} = O(1)$$

$$\text{deQueue} = O(4N) \quad : \quad \text{worst case}$$

Time complexity Queue Original ;

$$\text{enQueue} = O(1)$$

$$\text{deQueue} = O(N) \quad : \quad \text{worst case}$$

หากเปรียบเทียบจะเห็นได้ว่า queue ที่มาจาก stack ใช้เวลาในการแสดงผลมากกว่า 4 เท่า

Code : <https://ideone.com/CnYzHR>

```
1  #include<iostream>
2  #include<stack>
3  using namespace std;
4
5
6
7  struct Queue {
8      stack <int> s1, s2;
9      public:
10         void to_queue(stack <int> v)
11         {
12             if(!v.empty())
13             {
14                 s2.push(v.top());
15                 v.pop();
16                 to_queue(v);
17             }
18         }
19         void to_stack(stack <int> v)
20         {
21             if(!v.empty())
22             {
23                 s1.push(v.top());
24                 v.pop();
25                 to_stack(v);
26             }
27         }
28         void clear_s2()
29         {
30             if(!s2.empty())
31             {
32                 s2.pop();
33                 clear_s2();
34             }
35         }
36         void clear_s1()
37         {
38             if(!s1.empty())
39             {
40                 s1.pop();
41                 clear_s1();
42             }
43         }
44     }
```

```
44     void enqueue(int x)
45     {
46         s1.push(x);
47     }
48
49     int dequeue()
50     {
51         clear_s2();
52         to_queue(s1);
53         if (s2.empty()) {
54             cout << "Q is Empty";
55             exit(0);
56         }
57         int x = s2.top();
58         s2.pop();
59         clear_s1();
60         to_stack(s2);
61         return x;
62     }
63 };
64
65 int main()
66 {
67     Queue q;
68     q.enqueue(1);
69     q.enqueue(2);
70     q.enqueue(3);
71     q.enqueue(100);
72
73     cout << q.dequeue() << '\n';
74     cout << q.dequeue() << '\n';
75     cout << q.dequeue() << '\n';
76     cout << q.dequeue() << '\n';
77     cout << q.dequeue() << '\n';
78 }
```

# OUTPUT

```
1
2
3
100
Q is Empty
```