(1)

$$\underbrace{an}_{\text{add کردن n عنصر به صف}} + \underbrace{b(n+(n-1)+(n-۲)+\dots+۲+1)}_{\text{پیدا کردن بزرگترین عنصر صف}} + C$$

$$= an + b\left(\frac{n(n+1)}{۲}\right) + C = \frac{b}{۲}n^۲ + \left(a+\frac{b}{۲}\right)n + C \rightarrow O(n^۲)$$

(۲)

```
class Queue{

 int[]a;
  int first= -1
  int size=٠

    Queue(int size){
       a= new int[size];
       }

Queue~~(int size)~~() {super(100);}

void enqueue(int elemet){

       if(size<a.length){
       a[(first+size)% a.length]=element;
       size++;
       }


  }
```

Scanned with CamScanner

```
int deqaeue(){
 if (size>.) {
  int max = first;
  f(int i=1 ;i<size;i++)
    if(a[ (first+i) % a.length] >a[max] )
  max = (first +i) % a.length;
  int result = a[max];
  for ( int i=max ; i < max+( size-((max-first)%
                                              a.len+1)),
                   i++)
a [i %.a.len ]= a [(i+1) %.a.len];
  size--;
  return result;}
```

```
class TwinQueue {
  Element[]a; int first=-1; int size=0;
  TwinQueu(int n) {   a =new Element[n];
  }
TwinQueue() { super(100);}
 void enqueue( int n, int qunum ){
  if(size<a.len){
   a[(first+size)%a.len] =new Element(n,
                                    (Qunum==1?true:False
   size++                                           );
 }}
int deque(int qunum){
 if (size>0){ int target= first;
  wile(a[target].b!=(qunum==1)) target=(target+1)
                                              %a.len
 int resut = a[target.value;                      ;
 for(int i=target; i<target+(size-((target-first))%a.len+
      i++)                                        ;
a[i%a.len] = a[(i+1)%a.len];
 size--;     return resalt;}}}
```

```
class Element {
    int value;   boolean b;
    Element (int value, boolean b){
        this.value=value;   this.b=b; }}
```

۵ خیر طبق تعریف صف ، صف ترتیب را حفظ می‌کند می‌توان با استفا
ده ثابت کرد که k عمل enqueue ، k عمل dequeue انجام داده و آن ها را با
c و d نشان ~~ده~~ دهیم تمام حالت ها زیر با هم برابر است

۴ $(r-f+n)\%n$

Scanned with CamScanner