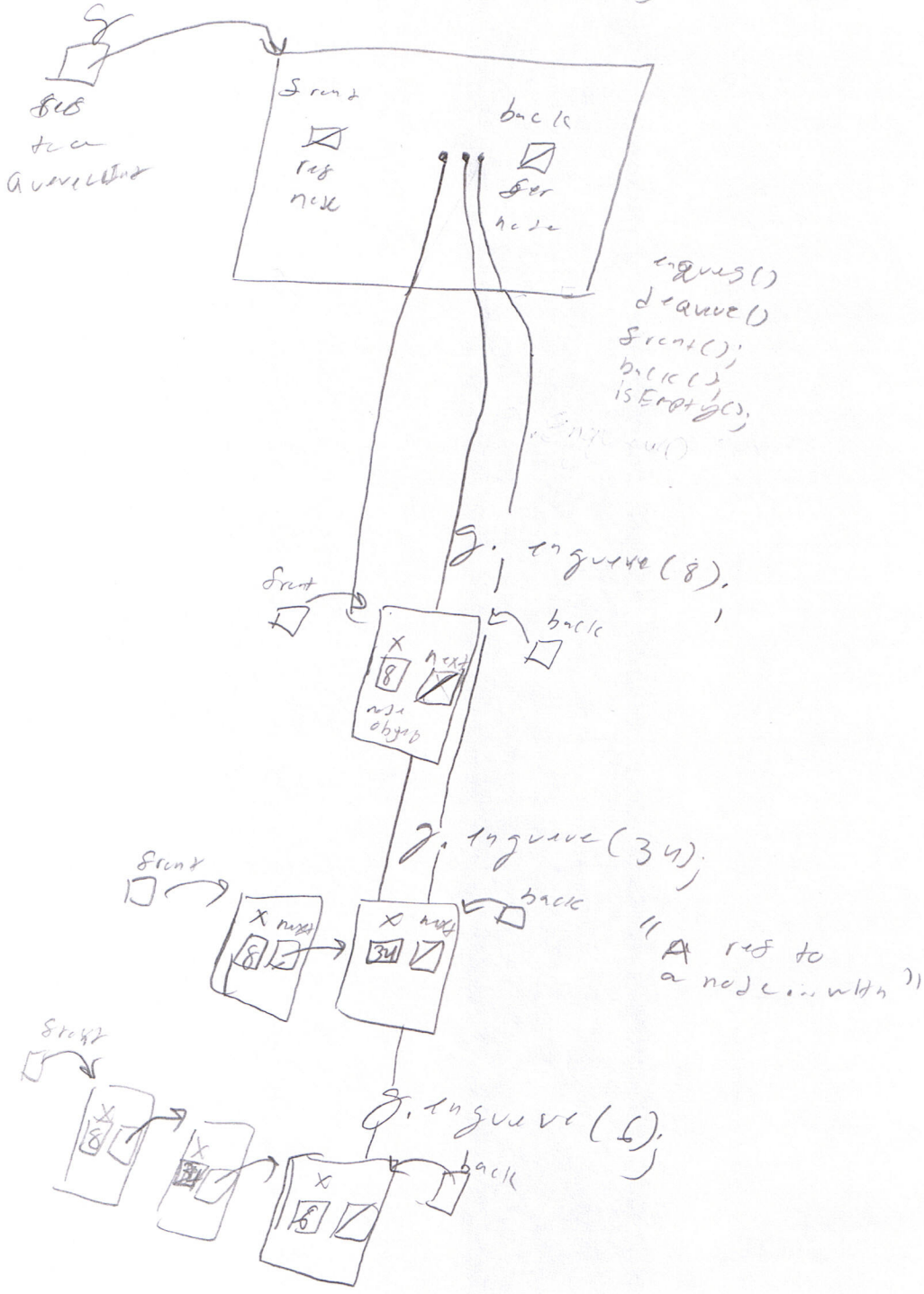
Cmp 338

9.22.09

QueueLLInt

s = new QueueLLInt();



Cmp 338

(2)

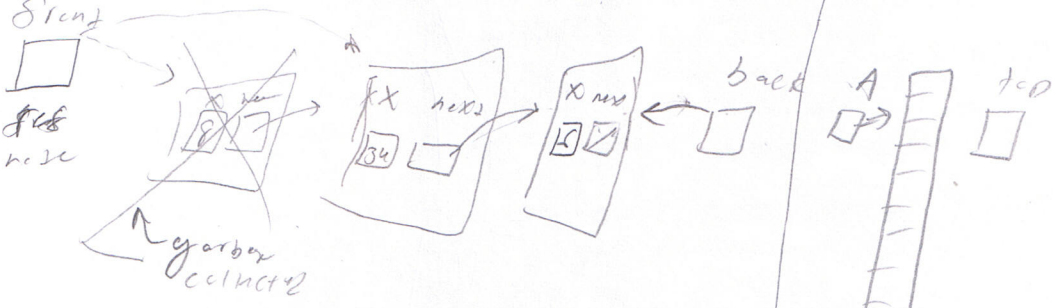
9.22.09

g.dequeue();

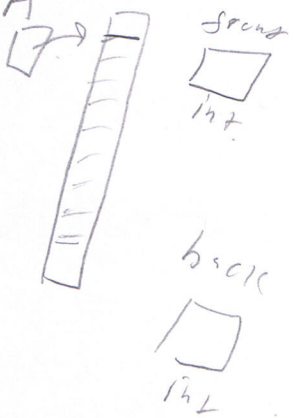
temp

8

create a Queue at time



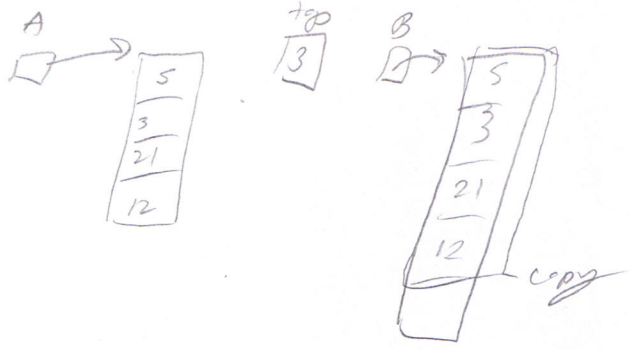
Making a Stack any size (made size)




```
public class StackInt
private int[] A;          constructor
public StackInt(int size)
{
    A = new int(size);
}
```

3

Why make the array very large (1000)




```
int[] B; new int[A.length+1]
for (int i=0; i<A.length; i++
    B(i) = A(i)
A = B;
```

```
public void push (int v)
{
  top++;
  if (top == A.length) resize();
  A[top] = v;
}
```

```
public void resize()
{
  int [] b = new int [A.length+1];
  for (int i=0; i<A.length; i++)
      B[i] = A[i]
  A = B;
}
```

Cmp 338

(4)

9.22.09

The Size By	Array size after each resize operation: (Copy counted)
1	1 2 3 4 5 6 7 8 9 10 + ... + 1023 = $\frac{1023(1024)}{2}$
Double	1 2 4 8 16 32 64 128 256 512 = 1023

$2n - 1$

$\frac{n(n+1)}{2}$

0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

n	$\frac{n^2+n}{2}$	$2n-1$
1000	500,500	1999
1,000,000	$\approx 10^{12}$	$2\times10^6-1$
10^9	$\approx 10^{18}$	$2\times10^9-1$

$A = 1 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{256} + \frac{1}{512}$

$512 * A$

$511 = 256 + 64 + 32 + 16 + 8 + 4 + 2 + 1$

$1 + 3 + 9 + 27 + 81 + 243 = \frac{1.5n - 1}{2n - 1} > \Theta(n)$

```
public void resize()
{
    int[] B = new int[A.length * 2];
    for (int i = 0; i < A.length; i++)
        B[i] = A[i];
    A = B;
}
```