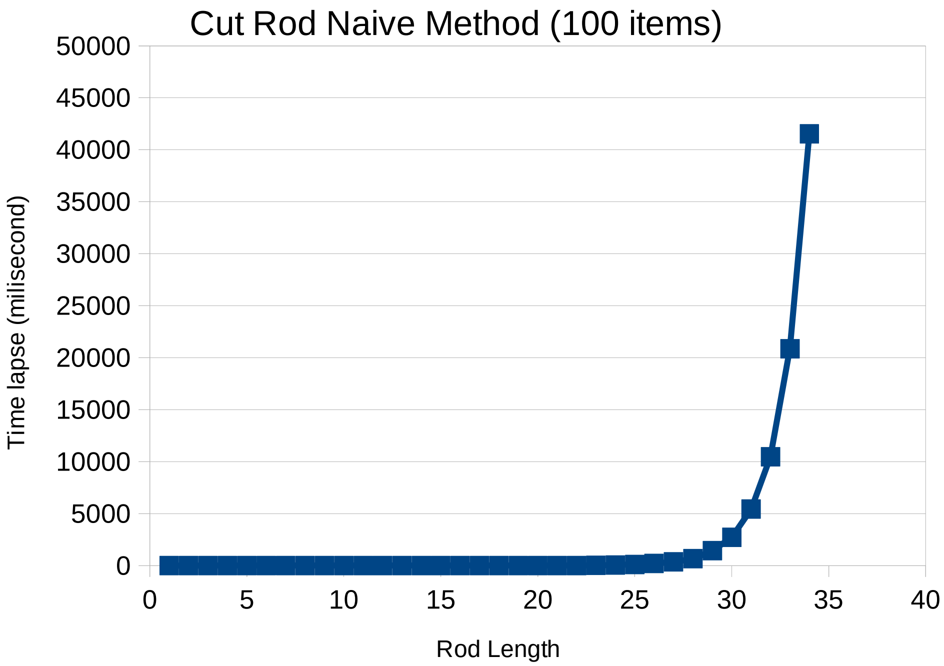
HW4
Aida Sharif Rohani

Array:
[4, 14, 20, 15, 6, 2, 11, 6, 11, 9, 17, 16, 13, 11, 1, 17, 20, 18, 11, 14, 8, 14, 18, 13, 9, 13, 10, 9, 4, 20, 12, 3, 10, 9, 1, 13, 9, 19, 11, 5, 1, 7, 16, 13, 20, 17, 8, 11, 5, 3, 14, 16, 8, 10, 3, 4, 3, 8, 15, 13, 9, 9, 18, 3, 19, 18, 6, 16, 4, 15, 15, 4, 3, 9, 4, 17, 8, 5, 2, 10, 4, 12, 11, 10, 10, 5, 11, 12, 18, 20, 19, 14, 14, 9, 20, 2, 8, 1, 16, 9]

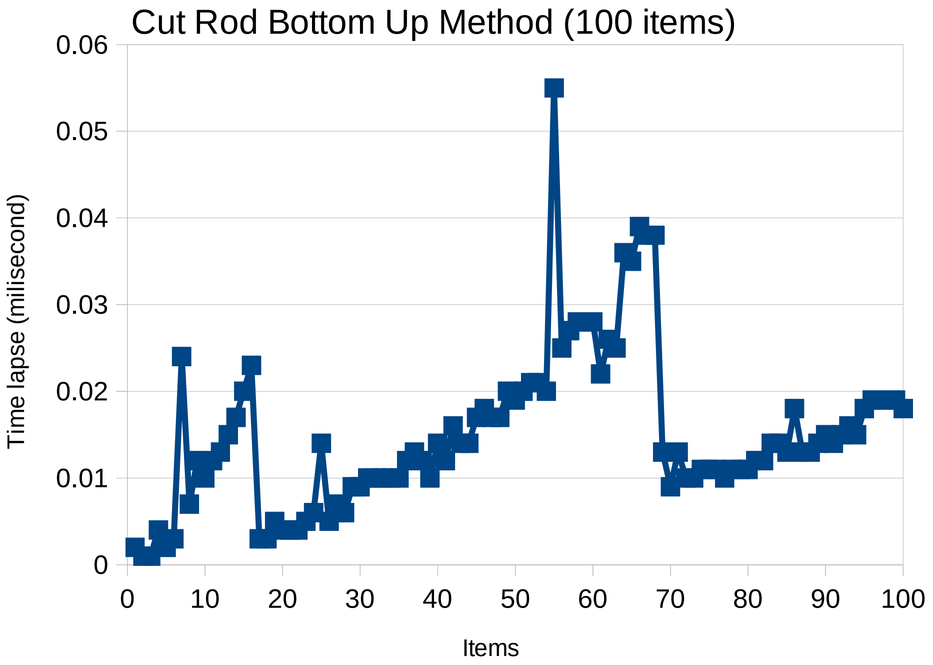
Result:

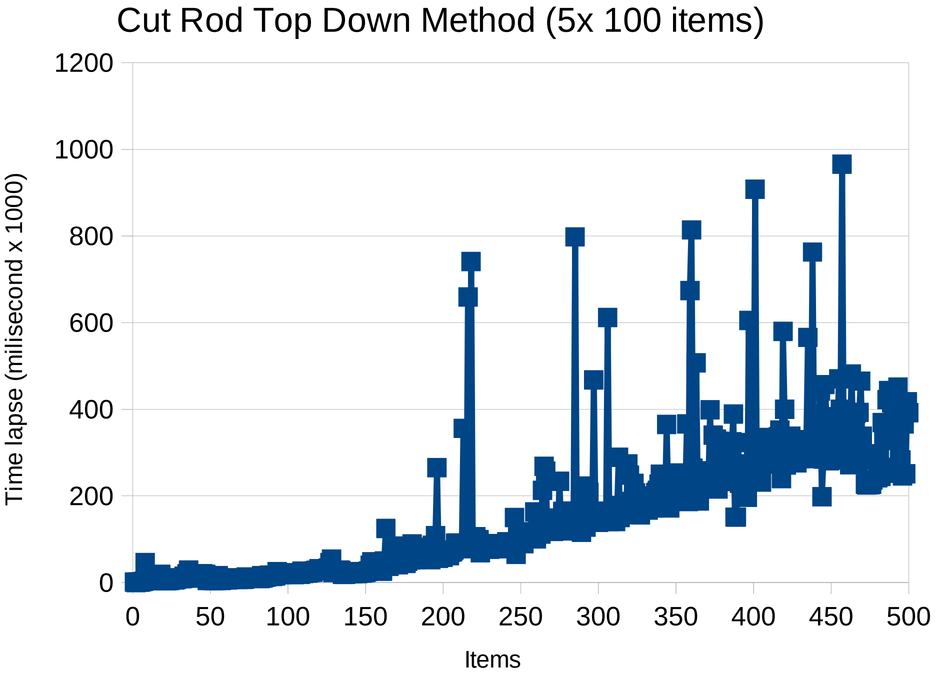
1- The maximum length calculated was 34. As the length increases the number of calculations exponentially increases. The shape of the graph is exponential, it means we can't calculate large array of elements with this method and it's very expensive and lengthy time wise.

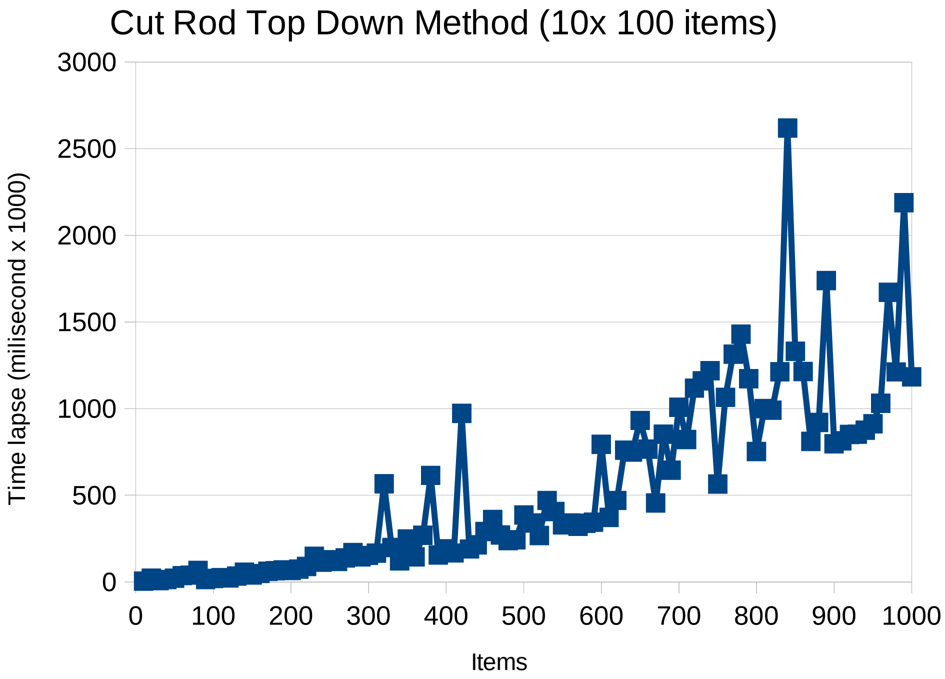


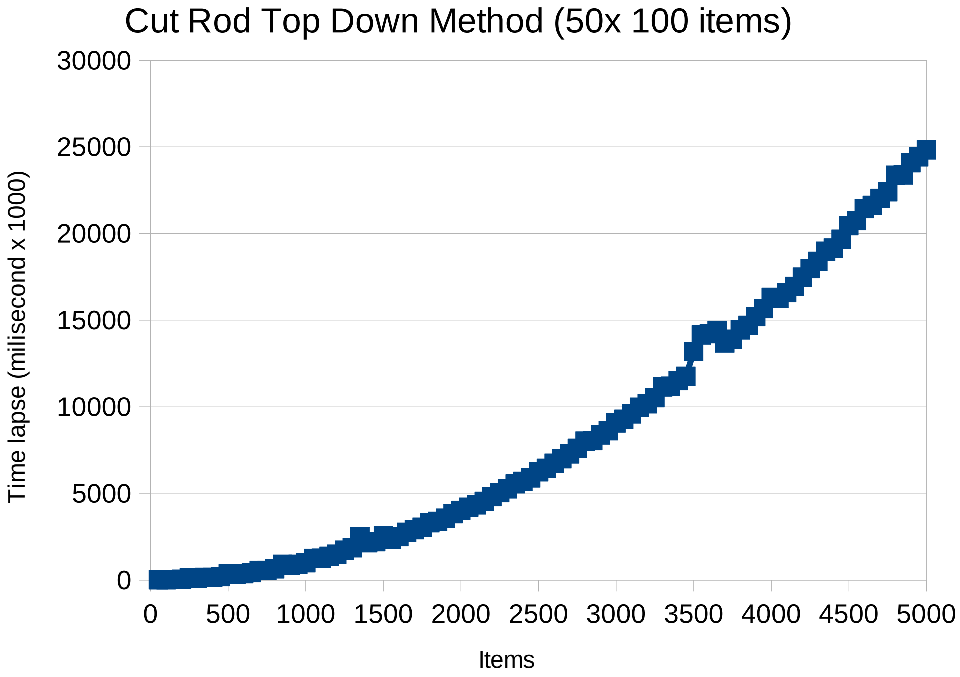
2- In top down in contrast to naive method all the numbers are calculated in less than 0.06 milliseconds. It shows that memoization makes it much faster to do the calculations as each number is only calculated once and after that it's saved in an array. The plot is not very smooth but the higher than average points are just noises and it's normal in order of milliseconds.

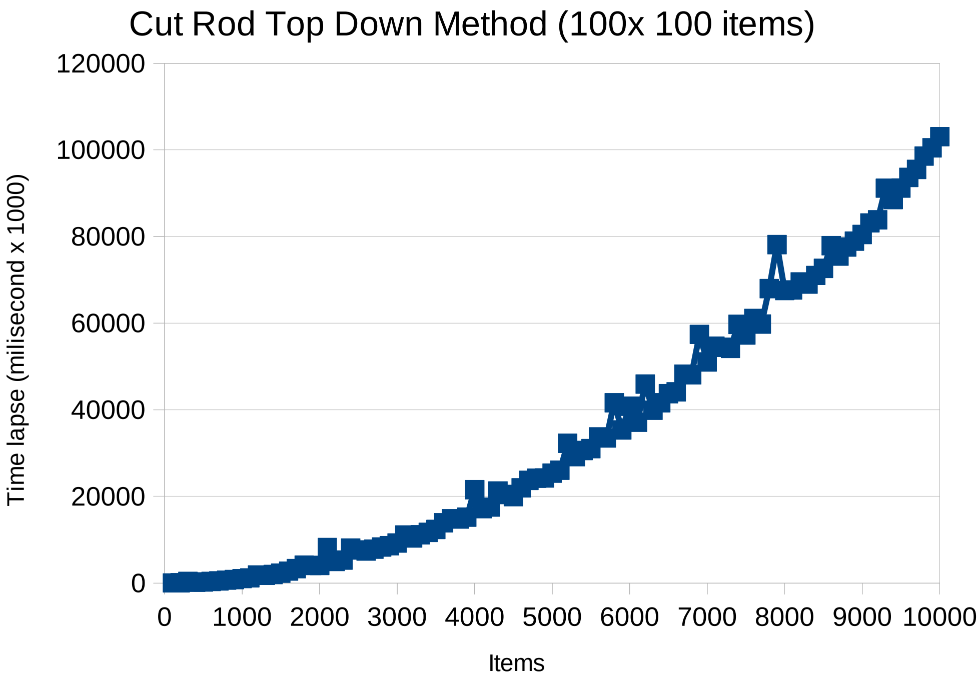
The higher elements makes the graph smoother and the time is increasing in n^2. For example when doing for 100x times the time increases by 10000 times. Which implies we can calculate much larger arrays compared to naive method.






Cut Rod Top Down Method (10x 100 items)
Time lapse (milisecond x 1000)
3000
2500
2000
1500
1000
500
0
0
100
200
300
400
500
600
700
800
900
1000
Items

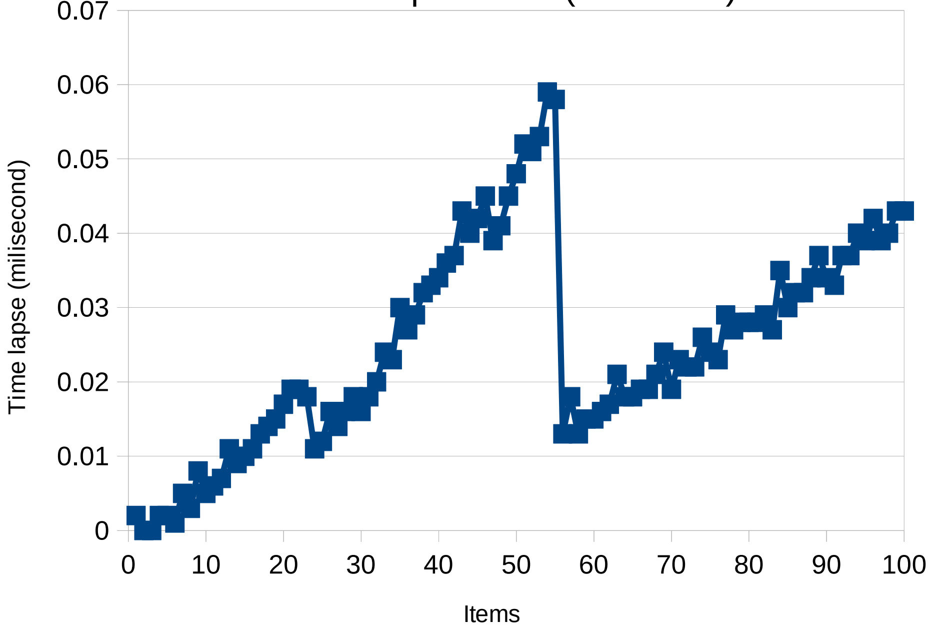

Cut Rod Top Down Method (50x 100 items)
Time lapse (milisecond x 1000)
30000
25000
20000
15000
10000
5000
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Items



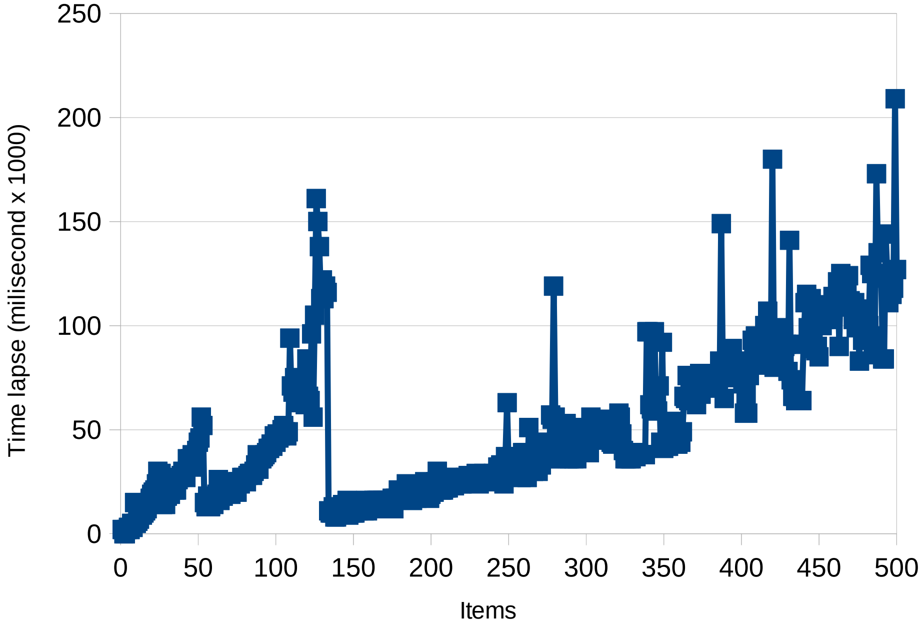
3-In bottom up method we get the same results as bottom up; again all the numbers are calculated in less than 60 milliseconds. It shows that bottom up methods makes it much faster to do the calculations as each number is only calculated once and after that it's saved in an array. The plot is not very smooth but the higher than average points are just noises and it's normal in order of milliseconds.

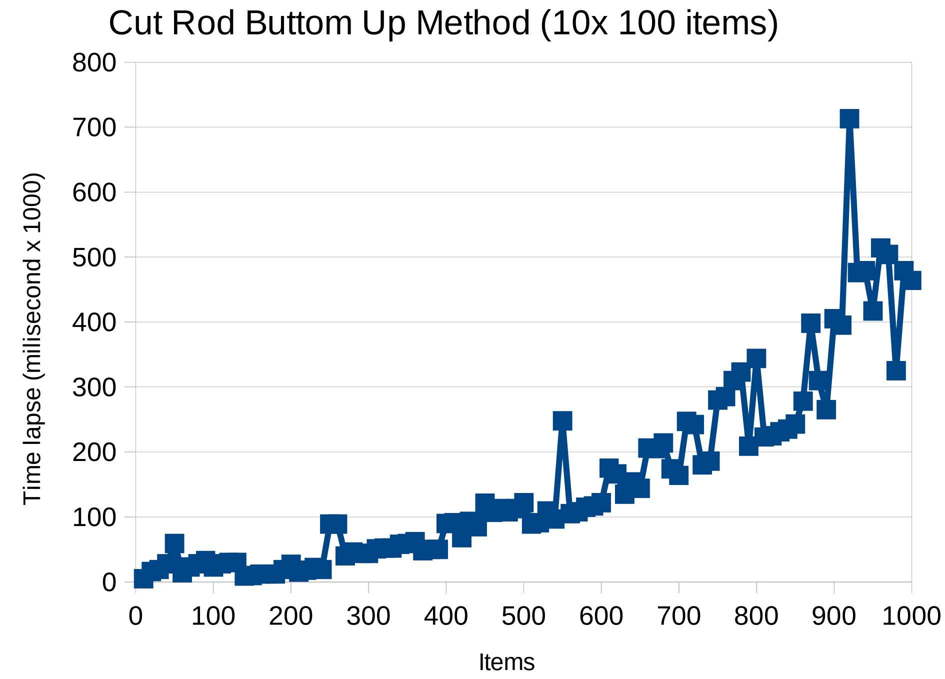
The higher elements makes the graph smoother and the time is increasing in n^2. For example when doing for 100x times the time increases by 10000 times. Which implies we can calculate much larger arrays compared to naive method.

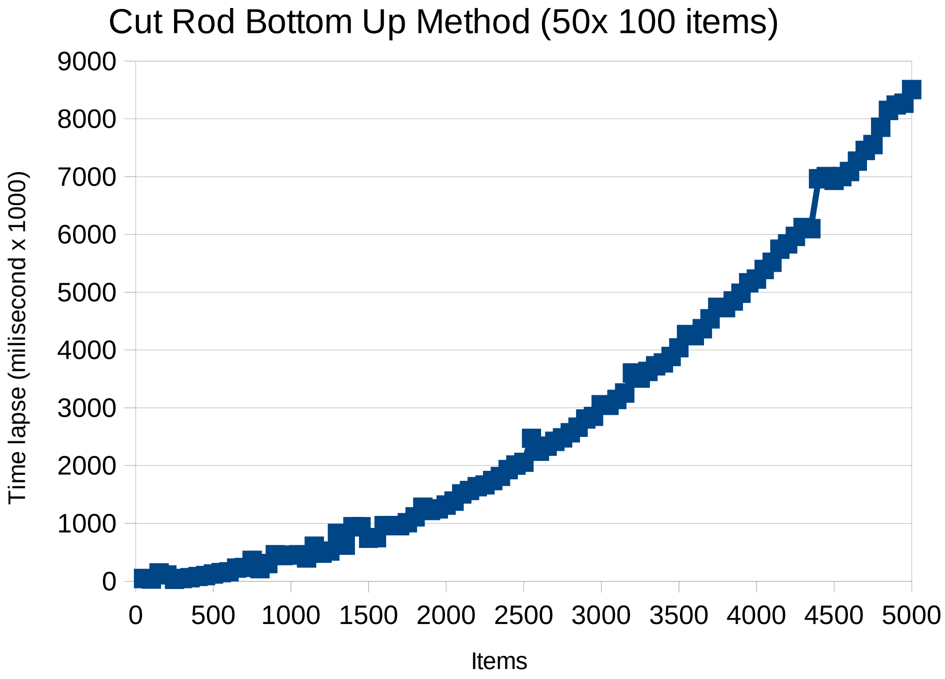
Cut Rod Bottom Up Method (100 items)

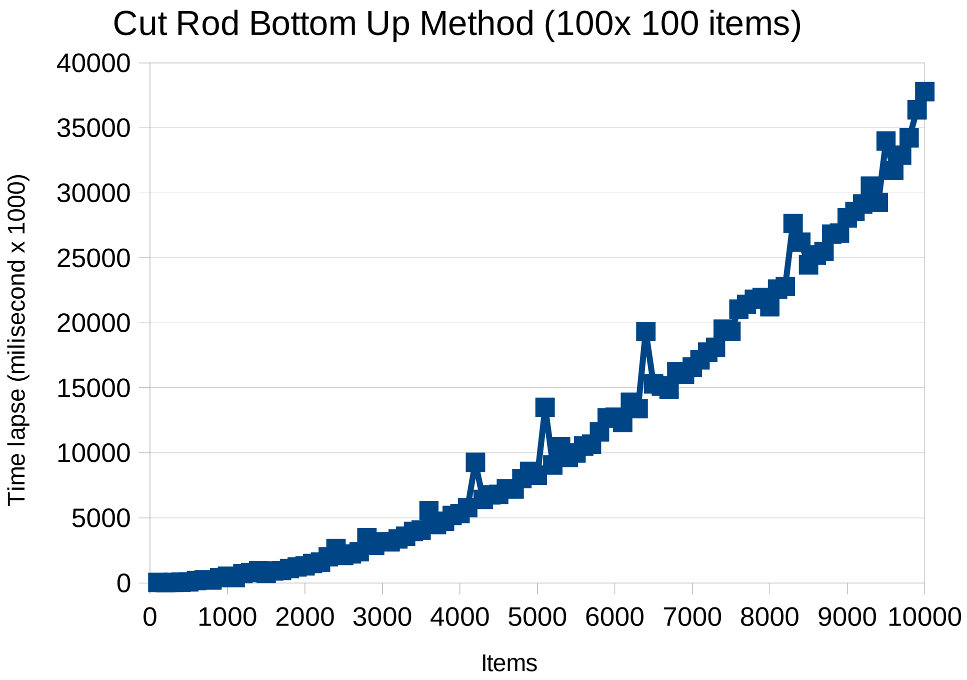


Cut Rod Buttom Up Method (5X 100 items)




Cut Rod Buttom Up Method (10x 100 items)
Time lapse (milisecond x 1000)
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000
Items


Cut Rod Bottom Up Method (50x 100 items)
Time lapse (milisecond x 1000)
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
Items


Cut Rod Bottom Up Method (100x 100 items)
Time lapse (milisecond x 1000)
40000
35000
30000
25000
20000
15000
10000
5000
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Items