```
import random as rn
import time
import matplotlib.pyplot as plt
```

Binary search trees

```
class rootNode:
    def __init__(self, key):
        self.left=None
        self.right=None
        self.val=key

def insertionNode(node,keyvalue):
    if node is None:
        return rootNode(keyvalue)
    else:
        if node.val==keyvalue:
            return node
        else:
            if node.val<keyvalue:
                node.right=insertionNode(node.right,keyvalue)
            else:
                node.left=insertionNode(node.left,keyvalue)
            return node

def inorder(root):
    if root:
        inorder(root.left)
        print(root.val,',',end='')
        inorder(root.right)

bst = rootNode(30)
bst=insertionNode(bst,50)
bst=insertionNode(bst,75)
bst=insertionNode(bst,80)
bst=insertionNode(bst,60)
bst=insertionNode(bst,65)
bst=insertionNode(bst,30)
bst=insertionNode(bst,100)
print("Inorder of binary search tree:")
inorder(bst)
```

```
Inorder of binary search tree:
30 ,50 ,60 ,65 ,75 ,80 ,100 ,
```

```
def inorderTraversal(root):
    if root:
        inorderTraversal(root.left)
        inorderTraversal(root.right)
```

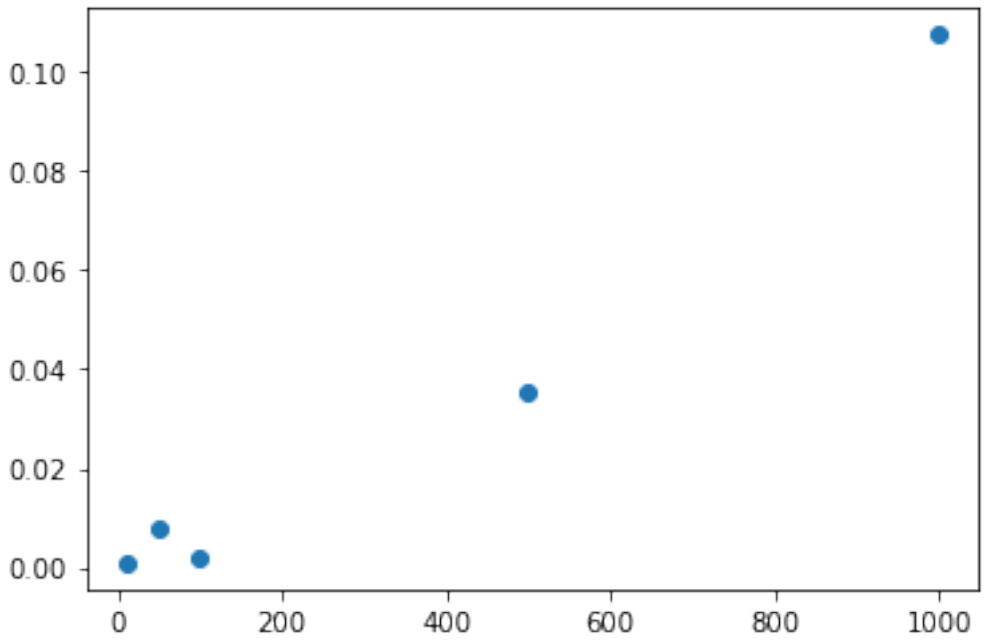
```
size=[10,50,100,500,1000]
bstList=['bst1','bst2','bst3','bst4','bst5','bst6']
runTime=[]
for i in range(len(size)):

    start=time.time()
    bstList[i]= rootNode(rn.randint(1,100000))

    for j in range(size[i]):
        bstlist[i]=insertionNode(bstList[i],rn.randint(1,100000))

    inorder(bstList[i])

    end=time.time()

    runTime.append(end-start)

plt.scatter(size,runTime)
plt.show()
```

```
8587 ,27599 ,27829 ,44052 ,44159 ,45875 ,55075 ,57360 ,75393 ,78376 ,9
4510 ,356 ,1419 ,2003 ,2058 ,4423 ,5372 ,7825 ,9079 ,9370 ,9730 ,9777
,13649 ,16383 ,20182 ,20826 ,21468 ,23418 ,23622 ,26430 ,29158 ,30336
,31250 ,34806 ,35216 ,37953 ,38689 ,41259 ,45149 ,45307 ,45527 ,49626
,50412 ,51516 ,54155 ,54422 ,54583 ,55706 ,61146 ,63013 ,71566 ,83582
,86682 ,89124 ,89492 ,90986 ,91285 ,91541 ,94741 ,95523 ,96328 ,98471
,142 ,145 ,1974 ,2389 ,2656 ,3724 ,4562 ,5645 ,7473 ,8409 ,8475 ,9036
,9391 ,10499 ,11330 ,11534 ,14959 ,16915 ,18437 ,19548 ,19619 ,20820 ,
21299 ,23853 ,24293 ,25299 ,25467 ,25548 ,25937 ,26030 ,26681 ,28366 ,
28604 ,29448 ,30165 ,30974 ,31694 ,31800 ,32036 ,32777 ,34316 ,34361 ,
36603 ,36635 ,37252 ,38554 ,38836 ,38885 ,39071 ,39773 ,41580 ,41889 ,
43602 ,44059 ,44750 ,44903 ,49033 ,49181 ,49304 ,49655 ,51008 ,51496 ,
52054 ,52546 ,53813 ,57071 ,57971 ,58011 ,58371 ,58637 ,64777 ,68056 ,
70312 ,71673 ,72809 ,73181 ,73547 ,74058 ,74600 ,74971 ,76322 ,76662 ,
76702 ,76773 ,78042 ,78636 ,78653 ,81030 ,81438 ,85252 ,86758 ,87082 ,
92449 ,93085 ,94380 ,94707 ,94991 ,96387 ,98131 ,98390 ,98887 ,50 ,181
,215 ,342 ,761 ,1046 ,1075 ,1112 ,1299 ,1369 ,1391 ,1398 ,1951 ,2103 ,
2436 ,2490 ,2646 ,3005 ,3256 ,3320 ,3556 ,3740 ,3743 ,3796 ,3893 ,4062
,4453 ,4683 ,4818 ,5820 ,5835 ,5965 ,6266 ,6633 ,6724 ,6853 ,6966 ,728
4 ,7313 ,7324 ,7420 ,7993 ,8867 ,8995 ,9153 ,9713 ,10133 ,10452 ,10639
,11601 ,11698 ,11908 ,12361 ,12376 ,12451 ,12581 ,13116 ,13527 ,13634
,13781 ,13926 ,13961 ,14135 ,14841 ,14846 ,15003 ,15208 ,15454 ,15476
,15756 ,15853 ,16076 ,16271 ,16614 ,16615 ,16719 ,16728 ,16978 ,17007
,17233 ,17430 ,17458 ,17668 ,17701 ,17744 ,18098 ,18464 ,18557 ,19583
,19704 ,20088 ,20536 ,20832 ,20896 ,20976 ,21100 ,21212 ,21276 ,21609
,22163 ,22202 ,22313 ,22654 ,22659 ,22807 ,23401 ,23669 ,23679 ,23802
,24010 ,24011 ,24658 ,24922 ,25170 ,25200 ,25284 ,25650 ,25713 ,26027
,26203 ,26381 ,26664 ,26865 ,26911 ,27076 ,27146 ,27335 ,27519 ,27648
,28107 ,28157 ,28244 ,28786 ,28850 ,29499 ,29529 ,29734 ,30181 ,30333
,30490 ,31323 ,31759 ,31793 ,31973 ,31974 ,32038 ,32056 ,32147 ,32588
```

,32664 ,32723 ,33191 ,33527 ,33923 ,34379 ,34552 ,34696 ,34976 ,34995
,35243 ,35295 ,35425 ,35635 ,35931 ,35967 ,36870 ,36888 ,36910 ,36957
,37030 ,37282 ,37350 ,37619 ,37623 ,37857 ,37980 ,38121 ,38277 ,38335
,38370 ,38442 ,38531 ,38628 ,38965 ,39257 ,39292 ,39548 ,39776 ,39973
,39983 ,40106 ,40404 ,40498 ,40779 ,40896 ,40938 ,41010 ,41524 ,42369
,42571 ,42754 ,42784 ,42936 ,43265 ,43637 ,43820 ,43895 ,43944 ,43948
,44071 ,44100 ,44518 ,44726 ,44824 ,44885 ,44952 ,45200 ,45562 ,45613
,45632 ,45682 ,45695 ,45835 ,46169 ,46346 ,46357 ,46554 ,47046 ,47269
,47578 ,47598 ,47829 ,48049 ,48052 ,48104 ,48212 ,48721 ,48901 ,49131
,49598 ,49687 ,49869 ,50063 ,50111 ,50115 ,50227 ,50626 ,50660 ,50887
,50957 ,51024 ,51149 ,51156 ,51469 ,51729 ,51829 ,51870 ,52280 ,52329
,52539 ,52790 ,53388 ,53498 ,53662 ,53804 ,53826 ,53829 ,53972 ,53974
,53995 ,54093 ,54239 ,54501 ,54505 ,54566 ,54567 ,54693 ,54928 ,55004
,55008 ,55063 ,55074 ,55076 ,55391 ,55611 ,55828 ,55897 ,55959 ,57093
,57360 ,57490 ,57950 ,58696 ,58717 ,58759 ,58807 ,58872 ,59154 ,59313
,59347 ,59694 ,59747 ,59749 ,59983 ,60193 ,60226 ,60600 ,60762 ,61486
,61885 ,61891 ,62694 ,62713 ,62973 ,63282 ,63543 ,63854 ,63947 ,64034
,64107 ,64139 ,64514 ,64566 ,64620 ,64637 ,64771 ,64802 ,64920 ,65003
,65146 ,65725 ,65892 ,66515 ,66547 ,66673 ,67229 ,67412 ,67414 ,67446
,68220 ,69281 ,69490 ,69515 ,69623 ,69800 ,69966 ,70031 ,70296 ,70367
,70499 ,70876 ,70965 ,72072 ,72076 ,72156 ,72301 ,72379 ,72512 ,72616
,72931 ,73379 ,73515 ,73824 ,73851 ,73969 ,74255 ,74499 ,74662 ,74730
,75282 ,75432 ,75479 ,75697 ,76438 ,76613 ,77183 ,77367 ,77511 ,77803
,77830 ,78053 ,78351 ,78447 ,78470 ,78555 ,78677 ,78831 ,79009 ,79172
,79311 ,79489 ,79742 ,79845 ,79879 ,80311 ,80497 ,80498 ,80587 ,80862
,81023 ,81176 ,81217 ,81371 ,81391 ,81557 ,81832 ,81889 ,82303 ,82307
,82503 ,82506 ,82766 ,83511 ,83847 ,83857 ,83954 ,84017 ,84035 ,84166
,84264 ,84368 ,84727 ,85108 ,85308 ,86003 ,86018 ,86135 ,86451 ,86768
,86815 ,86849 ,87100 ,87238 ,87240 ,87358 ,87595 ,87652 ,87691 ,87770
,87948 ,88379 ,88395 ,88397 ,88758 ,88795 ,88858 ,88989 ,89234 ,89429
,89942 ,89959 ,90260 ,90363 ,90639 ,90798 ,90871 ,90892 ,90980 ,91018
,91534 ,91593 ,91607 ,91932 ,91948 ,91984 ,92053 ,92128 ,92169 ,92301
,92439 ,92617 ,93200 ,93352 ,93362 ,93398 ,93636 ,93637 ,93928 ,93992
,94387 ,94609 ,95114 ,95118 ,95149 ,95184 ,95439 ,95866 ,97295 ,98363
,98388 ,98841 ,98862 ,98869 ,98945 ,98963 ,99061 ,99170 ,99747 ,99846
,99960 ,14 ,50 ,188 ,215 ,299 ,466 ,743 ,768 ,806 ,843 ,954 ,992 ,1202
,1225 ,1290 ,1345 ,1525 ,1577 ,2200 ,2236 ,2311 ,2363 ,2388 ,2393 ,254
2 ,2551 ,2654 ,2683 ,2736 ,2771 ,2784 ,2786 ,2863 ,2886 ,2915 ,2932 ,3
105 ,3129 ,3209 ,3252 ,3263 ,3527 ,3551 ,3767 ,3772 ,3784 ,3850 ,3887
,3980 ,4130 ,4204 ,4357 ,4394 ,4433 ,4435 ,4445 ,4461 ,4544 ,4559 ,457
9 ,4769 ,4773 ,4793 ,4818 ,5182 ,5263 ,5276 ,5336 ,5452 ,5517 ,5564 ,6
110 ,6126 ,6381 ,6458 ,6471 ,6485 ,6559 ,6594 ,6602 ,6615 ,6797 ,6954
,6983 ,6985 ,7123 ,7131 ,7419 ,7513 ,7609 ,7621 ,7637 ,7738 ,7756 ,776
3 ,7844 ,8090 ,8126 ,8279 ,8404 ,8507 ,8646 ,8668 ,8789 ,8808 ,8809 ,8
882 ,8905 ,8939 ,9126 ,9208 ,9248 ,9384 ,9452 ,9619 ,9655 ,9749 ,9977
,10070 ,10231 ,10258 ,10263 ,10307 ,10401 ,10459 ,10552 ,10754 ,10823
,11090 ,11322 ,11402 ,11430 ,11485 ,11820 ,11839 ,11849 ,11912 ,12013
,12180 ,12429 ,12478 ,12559 ,12598 ,12606 ,12687 ,12720 ,12902 ,13175
,13233 ,13238 ,13261 ,13290 ,13328 ,13331 ,13456 ,13527 ,13562 ,13716
,13864 ,14044 ,14068 ,14190 ,14207 ,14300 ,14639 ,14666 ,14674 ,14799

,14827 ,15027 ,15127 ,15377 ,15484 ,15509 ,15689 ,15692 ,15700 ,15846
,15869 ,15882 ,15959 ,16070 ,16373 ,16499 ,16541 ,16577 ,16706 ,16840
,17179 ,17397 ,17580 ,17622 ,17942 ,18175 ,18297 ,18343 ,18815 ,18822
,18943 ,18989 ,19064 ,19163 ,19266 ,19312 ,19316 ,19371 ,19400 ,19489
,19840 ,19848 ,20012 ,20095 ,20110 ,20126 ,20193 ,20254 ,20443 ,20483
,20553 ,20560 ,20640 ,20749 ,20752 ,20877 ,20905 ,21203 ,21292 ,21305
,21378 ,21484 ,21907 ,22167 ,22174 ,22196 ,22233 ,22442 ,22445 ,22833
,22895 ,23103 ,23247 ,23294 ,23421 ,23445 ,23524 ,23554 ,23579 ,23647
,23848 ,23864 ,23900 ,23915 ,24150 ,24159 ,24162 ,24229 ,24312 ,24568
,24573 ,24587 ,24595 ,24696 ,24745 ,24795 ,24862 ,24881 ,25053 ,25526
,25540 ,25613 ,25641 ,25862 ,26030 ,26062 ,26068 ,26150 ,26387 ,26507
,26610 ,26703 ,26856 ,27392 ,27475 ,27594 ,27632 ,27643 ,27665 ,27785
,27861 ,28077 ,28241 ,28260 ,28319 ,28490 ,28527 ,28968 ,29019 ,29052
,29245 ,29292 ,29319 ,29460 ,29521 ,29580 ,29596 ,29604 ,29651 ,29692
,29735 ,29799 ,29860 ,29870 ,29921 ,29940 ,30093 ,30140 ,30167 ,30222
,30228 ,30376 ,30443 ,30517 ,30621 ,30648 ,30741 ,30857 ,31015 ,31081
,31125 ,31142 ,31216 ,31230 ,31296 ,31438 ,31528 ,31658 ,32434 ,32499
,32572 ,32674 ,32687 ,32767 ,33289 ,33346 ,33446 ,33523 ,33543 ,33596
,33756 ,33806 ,33873 ,34145 ,34224 ,34357 ,34677 ,34860 ,34862 ,35017
,35257 ,35487 ,35687 ,36070 ,36074 ,36141 ,36163 ,36298 ,36365 ,36569
,36673 ,36703 ,36719 ,36731 ,36925 ,36990 ,37012 ,37031 ,37105 ,37274
,37422 ,37452 ,37549 ,37554 ,37563 ,37640 ,37644 ,37645 ,37836 ,37856
,37891 ,38324 ,38445 ,38548 ,38551 ,38583 ,38755 ,38760 ,39035 ,39094
,39127 ,39180 ,39301 ,39409 ,39565 ,39904 ,39993 ,40100 ,40139 ,40217
,40442 ,40456 ,40606 ,40621 ,40814 ,40944 ,41083 ,41189 ,41196 ,41452
,41487 ,41624 ,41678 ,41743 ,41744 ,41825 ,41911 ,41941 ,42064 ,42102
,42105 ,42146 ,42333 ,42402 ,42502 ,42538 ,42650 ,42893 ,42923 ,42976
,43131 ,43203 ,43211 ,43232 ,43599 ,43696 ,43786 ,43861 ,43994 ,43996
,44091 ,44228 ,44232 ,44259 ,44263 ,44288 ,44425 ,44619 ,44788 ,44800
,44976 ,44990 ,45087 ,45103 ,45125 ,45153 ,45537 ,45783 ,45800 ,45829
,45886 ,45911 ,45914 ,45938 ,46005 ,46046 ,46117 ,46154 ,46267 ,46291
,46462 ,46469 ,46630 ,46712 ,46930 ,46941 ,46957 ,47069 ,47536 ,47559
,47642 ,47708 ,47737 ,47804 ,47876 ,48052 ,48206 ,48255 ,48267 ,48345
,48347 ,48386 ,48714 ,48826 ,48929 ,49003 ,49021 ,49046 ,49103 ,49189
,49246 ,49256 ,49339 ,49351 ,49371 ,49391 ,49445 ,49675 ,49728 ,49793
,49840 ,50279 ,50302 ,50365 ,50564 ,50919 ,50964 ,51145 ,51168 ,51313
,51357 ,51377 ,51504 ,51528 ,51626 ,51635 ,51659 ,51887 ,52078 ,52230
,52276 ,52298 ,52416 ,52549 ,52587 ,52620 ,52749 ,52750 ,52937 ,53225
,53383 ,53412 ,53423 ,53461 ,53486 ,53508 ,53574 ,53610 ,53763 ,54359
,54419 ,54541 ,54633 ,54648 ,54730 ,54816 ,54942 ,55085 ,55126 ,55636
,55643 ,55708 ,55772 ,55927 ,56172 ,56340 ,56499 ,56645 ,56874 ,56923
,56931 ,56939 ,57032 ,57064 ,57161 ,57216 ,57281 ,57704 ,58190 ,58244
,58291 ,58350 ,58479 ,58608 ,58660 ,58681 ,58776 ,58778 ,58790 ,58794
,58811 ,58944 ,59042 ,59099 ,59102 ,59364 ,59472 ,59513 ,59917 ,59963
,60099 ,60119 ,60154 ,60166 ,60216 ,60239 ,60258 ,60347 ,60474 ,60523
,60541 ,60582 ,60724 ,60811 ,60841 ,60901 ,60925 ,60944 ,60994 ,61001
,61034 ,61039 ,61140 ,61274 ,61292 ,61317 ,61405 ,61483 ,61490 ,61551
,61640 ,61649 ,61800 ,61806 ,61827 ,61880 ,61981 ,62177 ,62325 ,62353
,62412 ,62427 ,62479 ,62787 ,62799 ,62930 ,62960 ,63090 ,63099 ,63289
,63347 ,63422 ,63484 ,63597 ,63655 ,63700 ,63761 ,63767 ,63775 ,63828

,63839 ,63945 ,64064 ,64100 ,64294 ,64332 ,64583 ,64619 ,64671 ,64815
,64822 ,64943 ,65035 ,65044 ,65237 ,65336 ,65646 ,65791 ,65841 ,65846
,65887 ,65917 ,65928 ,66017 ,66033 ,66156 ,66255 ,66400 ,66452 ,66508
,66537 ,66549 ,66784 ,66797 ,67124 ,67290 ,67360 ,67374 ,67480 ,67776
,67872 ,68038 ,68142 ,68191 ,68320 ,68450 ,68534 ,68537 ,68538 ,68562
,68593 ,68769 ,68798 ,68909 ,68910 ,69208 ,69222 ,69337 ,69413 ,69446
,69616 ,69724 ,69751 ,69889 ,69937 ,70124 ,70205 ,70263 ,70403 ,70690
,70935 ,71218 ,71227 ,71461 ,71550 ,71699 ,71730 ,71771 ,71948 ,72158
,72239 ,72585 ,72638 ,72904 ,73000 ,73034 ,73090 ,73100 ,73120 ,73646
,73669 ,73720 ,73928 ,73977 ,74042 ,74267 ,74588 ,74701 ,74799 ,74839
,75033 ,75100 ,75102 ,75296 ,75464 ,75491 ,75703 ,75892 ,75908 ,75915
,76104 ,76125 ,76189 ,76260 ,76277 ,76331 ,76509 ,76557 ,76638 ,77151
,77201 ,77225 ,77228 ,77307 ,77490 ,77577 ,77667 ,78159 ,78273 ,78427
,78794 ,79018 ,79077 ,79084 ,79206 ,79245 ,79367 ,79568 ,79593 ,79670
,79716 ,79829 ,79909 ,80058 ,80131 ,80160 ,80359 ,80417 ,80457 ,80458
,80585 ,80588 ,80632 ,80696 ,80755 ,80821 ,80847 ,80863 ,80923 ,80957
,81130 ,81160 ,81309 ,81317 ,81420 ,81506 ,81685 ,81707 ,81803 ,82173
,82359 ,82521 ,82534 ,82664 ,82691 ,82822 ,82823 ,82847 ,82882 ,82916
,83082 ,83111 ,83465 ,83577 ,83580 ,84268 ,84287 ,84460 ,84506 ,84520
,84674 ,84900 ,85077 ,85141 ,85157 ,85301 ,85320 ,85530 ,85616 ,85643
,85734 ,85915 ,86016 ,86019 ,86253 ,86321 ,86459 ,86466 ,86600 ,86619
,86785 ,87055 ,87205 ,87216 ,87385 ,87389 ,87472 ,87542 ,87577 ,87609
,87644 ,87677 ,87885 ,88066 ,88118 ,88247 ,88393 ,88537 ,88597 ,88619
,88868 ,88998 ,89007 ,89018 ,89019 ,89040 ,89063 ,89064 ,89370 ,89541
,89596 ,89616 ,89705 ,89785 ,89800 ,89829 ,89869 ,89992 ,90091 ,90107
,90364 ,90369 ,90424 ,90472 ,90550 ,90807 ,91201 ,91260 ,91473 ,91482
,91499 ,91561 ,91605 ,91659 ,91680 ,91698 ,91765 ,92121 ,92185 ,92225
,92390 ,92449 ,92487 ,93077 ,93122 ,93291 ,93362 ,93668 ,93777 ,93786
,93802 ,93818 ,93825 ,94389 ,94681 ,94783 ,94811 ,94884 ,94886 ,94964
,94983 ,95023 ,95250 ,95359 ,95458 ,95525 ,95706 ,95781 ,95877 ,95936
,95975 ,96258 ,96490 ,96598 ,96635 ,96802 ,96943 ,97237 ,97575 ,97667
,97713 ,98012 ,98030 ,98386 ,98395 ,98503 ,98542 ,98567 ,98614 ,98830
,98861 ,98932 ,99021 ,99150 ,99519 ,99780 ,



Time Complexity

1) Time complexity for adding n elements is O(n*logn)

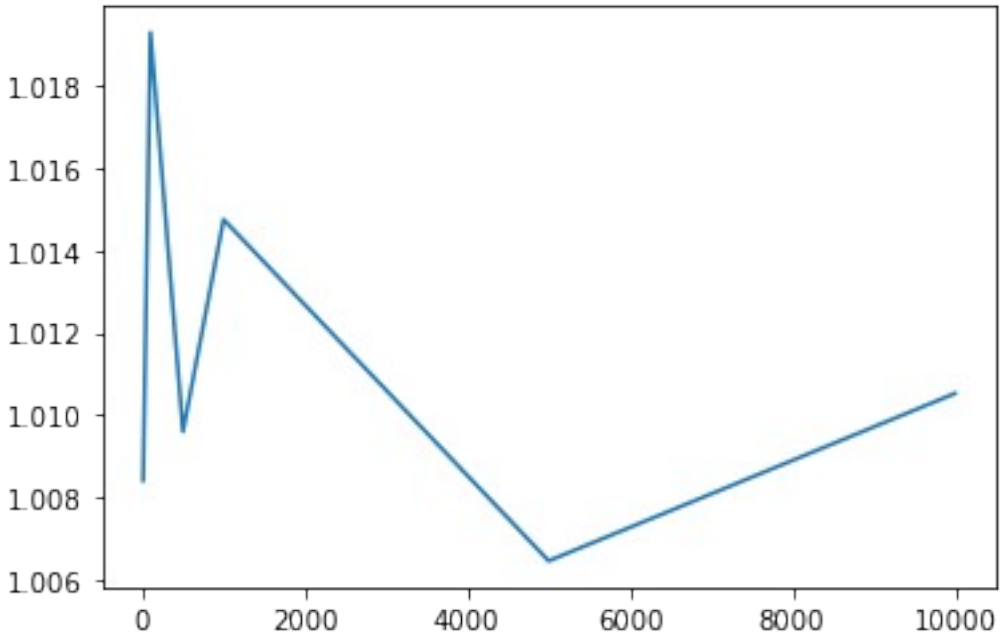
2) The average time complexity for tree sort is O(n*logn)

3) the worst case time complexity for tree sort is O(n^2)

Heapsort

```
def heapify(arr,n,i):

    keyMax=i
    leftchild=2*i+1
    rightchild=2*i+2
    temp=0

    if leftchild<n and arr[keyMax]<arr[leftchild]:
        keyMax=leftchild

    if rightchild<n and arr[keyMax]<arr[rightchild]:
        keyMax=rightchild

    if keyMax!=i:
        temp=arr[i]
        arr[i]=arr[keyMax]
        arr[keyMax]=temp
        heapify(arr,n,keyMax)
```

```
def maxHeap(arr):

    n=len(arr)
    a=n//2-1

    for i in range(a,-1,-1):
        heapify(arr,n,i)

def heapSort(arr):

    for i in range(len(arr)-1,0,-1):
        temp=arr[i]
        arr[i]=arr[0]
        arr[0]=temp1
        heapify(arr,i,0)

T=[]
n=[10,100,500,1000,5000,10000]

for i in range(len(n)):
    for i in range(n[i]):
        array=[]
        array.append(rn.randint(1,1000000))

    start=time.time()
    maxHeap(array)
    heapSort(array)

    time.sleep(1)
    end=time.time()
    T.append(end-start)

plt.plot(n,T)
plt.show()
```



(Here the time taken is 1s more than what is depicted in the graph)

Time Complexity:

1) Time complexity of heap sort is O(n)