```
import time
import matplotlib.pyplot as plt
#functio to perform binary search

def binary_serach(arr, low, high, key):
    while low <= high:
        mid = low + (high - low) // 2
        if arr[mid] == key:
            return mid # if key is found, return the index
        if arr[mid] < key:
            low = mid + 1
        else:
            high = mid - 1
    return -1

#main function
def main():
    n_values = []
    times = []

    r = int(input("Enter the number of runs:"))
    for _ in range(r):
        n = int(input("Enter the number of elements: "))
        arr = sorted(list(map(int, input("\n Enter the elements of an array: ").split())))
        key = int(input("\n Enter the key elements to be searched:"))
        repeat = 1000

        start = time.time()
        for _ in range(repeat):
            result = binary_serach(arr, 0, n - 1, key)
        end = time.time()

        if result != -1:
            print(f"key {key} found at position {result}")
        else:
            print(f"key {key} not found ")

        time_taken = (end - start) * 1000
```

```
        print(f"time taken to search a key element = {time_taken} mill seconds\n")

        n_values.append(n)
        times.append(time_taken)

        plt.figure()
        plt.plot(n_values, times, 'o-')
        plt.xlabel('Number of elements (n)')
        plt.ylabel('time taken (mill seconds)')
        plt.title('Binary Search Time Complexity')
        plt.grid(True)
        plt.show()

if __name__ == "__main__":
    main()
```