Please submit a pdf copy (at most 3 pages) of your own solutions to Problems 1 and 2, at SUCourse+ before February 24 (Monday), 23:30.

Problem 1 (20 points) Give an asymptotic tight bound for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. No explanation is needed.

(a) $T(n) = 2T(n/2) + n^3$

(b) $T(n) = 7T(n/2) + n^2$

(c) $T(n) = 2T(n/4) + \sqrt{n}$

(d) $T(n) = T(n-1) + n$

Problem 2 (55 points) Consider the following two algorithms for finding the n'th Fibonacci number. Note that the algorithm on the left hand side is recursive while the other one is iterative.

```
def recfib(n):
  if n < 3:
    return 1
  else:
    return recfib(n-1)+recfib(n-2)
```

```
def iterfib(n):
  cur, pre = 1, 1
  for i in range(n-2):
    cur, pre = cur+pre, cur
  return curr
```

(a) For each algorithm, determine its scalability theoretically by finding its best asymptotic worst-case running time. Please explain your answers by explicitly writing down the running time functions, solving recurrences if needed, and calculating the total cost.

(b) Implement these two algorithms using Python. For each algorithm, try to determine its scalability experimentally by running your programs with at least 15 different numbers n.

(i) To represent your experimental results, fill in the following table with the running times in seconds.

Algorithm	$n = 5$	$n = 10$	$n = 15$	...	...	...	...	...	...	...	...	...	...	...	...
Iterative	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Recursive	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Specify the properties of the machine (CPU, RAM, OS) where you run your programs.

(ii) To better observe the scalability of the algorithms with respect to these experimental results, plot them in a graph so that the x axis denotes the values of n and y axis denotes the CPU time in seconds.

(iii) Then please discuss the scalability of the algorithms with respect to your observations over the experimental results. In particular, try to answer the following questions: How does the computation time change as the input size increases? Do these experimental results confirm the theoretical results you found?