YSC2229: Assignment 1

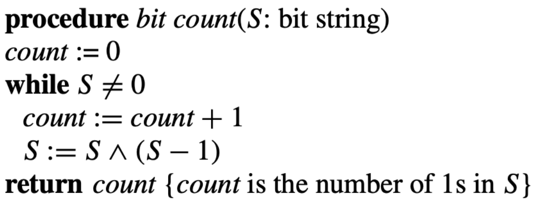
Deadline: February 11, 2021 at 5pm

(Strict Deadline: No Late Submission Accepted)

Submit your work in the jupyter-notebook format via Canvas by the deadline. Include your non-code answers (in pdf) on your submitted jupyter notebook. One should be able to see all your answers on the same jupyter notebook. This is an individual assignment. No collaboration with anyone is allowed.

Part 1:

1. Consider this pseudocode:



```
procedure bit count($S$: bit string)
count := 0
while $S \neq 0$
  count := count + 1
  $S := S \wedge (S - 1)$
return count {count is the number of 1s in $S$}
```

   where $\wedge$ means the bitwise AND operation. Bit stands for binary digit (= 0/1). Bit-string means an array of bits (or a string of bits), e.g., $S = 001110001$. For this task: (1) write the code in python, (2) calculate its $T(n)$, where n is the length of S, (3) show the big-O notation, (4) prove your big-O notation.

2. Given a set of the parameters of a polynomial function, write a program to compute the output of the polynomial function. You are not allowed to use the pow function or any available function for computing powers (you must create your own power function). For this task: (1) write a pseudocode, (2) write a program in python based on the pseudocode, (3) calculate its $T(n)$, (4) show the big-O notation, (5) prove your big-O notation.

Part 2:

1. Rank the following runtime costs based on their efficiency (most efficient ones come first). You must provide the proofs.

   (a) n^2

   (b) $\log n$

   (c) 1,000,000

   (d) n

   (e) 2^n

   (f) $n \log n$

2. Prove or disprove: $2^{2n} = O(2^n)$. Note, you must include the conditions if you prove it true .