CS 310

Test 1

Jingbo Wang

Answer 1(b): I would prefer to use program g because its output growth slower when the input size is big.

Answer 2: In order to prove this, we must find c and n_0 such that

$$2\text{n} + 5 \leq cn$$
$$2+\frac{5}{n} \leq c$$
$$\text{When } \ n_o = 1, \ (2+\frac{5}{n}) \ \text{ have max value}$$
$$\text{Max}(2+\frac{5}{n}) = 7$$
$$7 \leq c$$
$$\text{So, } \ \text{c} = 7$$
$$\text{Therefore, } \ \text{T(n)} \in O(n), when\ c = 7,\text{n}_0 = 1$$

Answer 3: The approximate base-2 logarithm of 500,000 is 2^{19}. This is derived as follows:

$$\begin{aligned} 500000 &= 500 \times 10^3 \\ &= 2^9 \times 2^{10} \\ &= 2^{19} \end{aligned}$$

Answer 4(a): Here, f is an upper bound of g because n! growth faster than 2^n when input size is huge.

Answer 4(b): Here, f is an upper bound of g because n^3 is lager than n^2 when input size is huge.

Answer 4(c): Here, g is an upper bound of f because $\log 2n = \log 2 + \log n < \log n$.

Answer 5: We consider the input size to be foo. The operations are counted as follows.

- Line 3: one operation regardless of the input size
- Line 8: a vector of a specific size can be allocated in one operation
- Line 9: one operation, an array element assignment
- Line 10: one operation, an array element assignment
- Line 11: for loop statement, runs $n-1$ times and one last time, two operations for each times $2(n-1)+2 = 2n$ operations.
- Line 10: four operations for each, it runs $n-1$ times, so $4(n-1) = 4n-4$ operations

Adding up all the values, we get the number of operations;

When is best case, it means Line 3 is true, and it return 0:

$$\mathrm{T(n)} = 1$$

When is worst case, it means Line 3 is false, and it return 0:

$$\begin{aligned} T(n) &= 1 + 1 + 1 + 1 + 2n + (4n - 4) \\ &= 6n \end{aligned}$$

Therefore, we conclude that the overall analysis is

$$\begin{aligned} T(n) &\in O(n) \\ &\in \Omega(1) \end{aligned}$$