CS 310

Assignment 0820

19 August 2020

Zhihao Jin(Bill)

Problem 1 — Algorithms typically have the following properties:

- the steps are stated *unambiguously* so that there is no question how a step is performed
- the algorithm is *deterministic* so that repeating the algorithm on the same input produces the same output
- the algorithm is *finite* because it terminates after a finite number of steps have been performed
- the algorithm produces *correct* output for a given input

For the following algorithm, for each property listed above, determine whether the algorithm exhibits this property:

```
unsigned max3(unsigned a, unsigned b, unsigned c)
{
  unsigned result = a;
  if (b > result)
  {
    result = b;
  }
  if (c > result)
  {
    result = c;
  }
  return result;
}
```

Answer: This algorithm is unambiguous because the syntax for the operations is well-understood. It is deterministic because it always produces the same output for a given input. It is finite because the number of lines of code executed (including the header) is strictly between 3 and 7 inclusive. It is correct because for all possible combinations of input it produces the correct output.

Problem 2 — Using the same properties as stated in problem 1, analyze the following algorithm:

```
void shuffle(vector<unsigned>& a)
{
  for (size_t index = 0; index < a.size(); index++)
  {
      swap(a.at(index), a.at(get_random_within(index, a.size()));
  }
}
```

Answer: This algorithm is unambiguous and quit easy to understand. However, it is not deterministic since the output will not be always same based on the random function. It is finite because the number of iterations is from zero to the size of vector. But it doesn't output anything, instead, swapping the position of the elements in the vector.

Problem 3 — What is the hexadecimal representation of 770_{10}?

Answer: The first few powers of 16 are:

$$
\begin{aligned}
16^0 &= 1 \\
16^1 &= 16 \\
16^2 &= 256 \\
16^3 &= 4096
\end{aligned}
$$

Thus we have:

$$
\begin{array}{r}
770 \\
\underline{-3 \times 256 = 768} \\
2 \\
\underline{-0 \times 16 = 0} \\
2 \\
\underline{-2 \times 1 = 2} \\
0
\end{array}
$$

And thus we have $770_{10} = 302_{16}$.

Problem 4 — Based on your answer to the previous problem, what is the binary representation of 768_{10}?

Answer: The first few powers of 2 are:

$$
\begin{aligned}
2^0 &= 1 \\
2^1 &= 2 \\
2^2 &= 4 \\
2^3 &= 8 \\
2^4 &= 16 \\
2^5 &= 32 \\
2^6 &= 64 \\
2^7 &= 128 \\
2^8 &= 256 \\
2^9 &= 512 \\
2^{10} &= 1024
\end{aligned}
$$

Thus we have:

$$\begin{array}{r}
768 \\
\underline{-1 \times 512 = 512} \\
256 \\
\underline{-1 \times 256 = 256} \\
0 \\
\underline{-0 \times 128 = 0} \\
0 \\
\underline{-0 \times 64 = 0} \\
0 \\
\underline{-0 \times 32 = 0} \\
0 \\
\underline{-0 \times 16 = 0} \\
0 \\
\underline{-0 \times 8 = 0} \\
0 \\
\underline{-0 \times 4 = 0} \\
0 \\
\underline{-0 \times 2 = 0} \\
0 \\
\underline{-0 \times 1 = 0} \\
0
\end{array}$$

And thus we have $768_{10} = 1100000000_2$.

Problem 5 — What is the decimal representation of `0x1abc`?

Answer: Based on the table below we can find that:

Hex number:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
decimal number:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

$$1 \times 16^3 + 10 \times 16^2 + 11 \times 16^1 + 12 \times 16^0 = 6844_{10}$$

And therefore, the decimal representation of `0x1abc` is 6844.