The Chinese University of Hong Kong
Department of Systems Engineering and Engineering Management

CSCI-2100-B Data Structures

2013-14 Assignment 2

1 General Information

The second written assignment accounts for 25% of the unit's assessment, and is due at 5:00pm on March 31, 2014.[1] You must submit to the assignment box **B08**, 5th floor, William M.W. Mong Engineering Building. You need to put down your student id and your name. **Late submission is not allowed**.

2 Questions

■ **Question** 1: The problem of Hailstone Numbers is to obtain a sequence of numbers $H_n = (n_1, n_2, \cdots)$ starting from a given positive number, n. Let $n_0 = n$ to start, repeat the following until $n_i = 1$: if n_i is an even number, n_{i+1} will be $n_i/2$, and if n_i is an odd number, n_{i+1} will be $3n_i + 1$. For example, given $n = 10$, $H_n = (5, 16, 8, 4, 2, 1)$. Write `C`-like code (or pseudo code in English). First, you have to write a function that takes n as input and output the sequence H_n. Second, you have to explore whether H_n can be obtained for every positive number n that is allowed in your computer. By `C`-like code, we mean it looks like `C` code, but we ignore syntax errors if you make your point clear. An interesting thing is for any positive number n it will find an H_n in a finite step. But, in theory, no one can prove it. *Tell us if you find one that cannot stop.*

[1] Departmental Guideline for Plagiarism (Department of Systems Engineering and Engineering Management): If a student is found plagiarizing, his/her case will be reported to the Department Examination Panel. If the case is proven after deliberation, the student will automatically fail the course in which he/she committed plagiarism. The definition of plagiarism includes copying of the whole or parts of written assignments, programming exercises, reports, quiz papers, mid-term examinations and final examinations. The penalty will apply to both the one who copies the work and the one whose work is being copied, unless the latter can prove his/her work has been copied unwittingly. Furthermore, inclusion of others' works or results without citation in assignments and reports is also regarded as plagiarism with similar penalty to the offender. A student caught plagiarizing during tests or examinations will be reported to the Faculty office and appropriate disciplinary authorities for further action, in addition to failing the course.

■ **Question** 2: Big-Oh notation is used for the worst-case analysis. We discussed the notation in `ch1.pptx`. The maximum rule is given in slide 1-46 in `ch1.pptx`. Following the maximum rule step-by-step, show the time complexity for "$g(n) = n^3 + 2n^2 + 10n$".

- let $g(n) = h(n) + k(n)$ where $h(n) = n^3 + 10n$ and $k(n) = 2n^2$, and then let $h(n) = h_1(n) + h_2(n)$ where $h_1(n) = n^3$ and $h_2(n) = 10n$.

First, use the maximum rule to find the Big-Oh notation for $h(n)$, and then use the maximum rule to find the Big-Oh notation for $g(n)$. You are required to provide all concrete constant numbers that are needed in Big-Oh notation. What does the maximum rule tell us to find a Big-Oh notation for a function?

■ **Question** 3: In `ch2.pptx`, we discuss the polynomial ADT (starting from the slide 2-15). The final goal is to be able to compute, for example, $h(x)+g(x)$ or $h(x) \times g(x)$, based on the ADT defined. It is important to note that we have to be able to maintain any polynomials in a data structure, in order to process them.

The space consumption: Consider Data-Structure-2 (slide 2-19 – 2-21), and Data-Structure-4 (slide 2-24 – 2-25). Here, the degree of a polynomial is denoted as n (refer to slide 2-18). Answer the following questions based on the definition of data structures.

- Q3a: How many bytes are needed for a term in a polynomial with the degree n for the two data structures respectively?
- Q3b: How many bytes are needed for a polynomial with the degree n for the two data structures respectively?
- Q3c: Which one is better?

Next, consider the following cases.

- The number of terms in a polynomial can vary. The maximum number of terms in a polynomial is $n + 1$. For example, the polynomial "$h(x) = 10x^4+20x^3+x^2+x+1$" has every possible term, where as "$g(x) = 10x^4+1$" has two terms. Let's call the terms that appear in a polynomial non-empty term.
  - Assume there are two sub-cases, 80% non-empty and and 20% non-empty. Given the two statistics, answer the three questions Q3a, Q3b, and Q3c.
- Some polynomials only have a few non-empty terms. Some may have at the two ends, for example, $h(x) = x^{999} + 10$, and some may have all non-empty terms in a relative close positions, for example, $g(x) = x^{99}+579x^{95}$.
  - In this case, answer the three questions Q3a, Q3b, and Q3c.

Finally, given all these possibilities, which of the two data structures will you choose? State your reasons.

■ **Question** 4: The Tower of Hanoi is a mathematical game or puzzle. Read `http://en.wikipedia.org/wiki/Tower_of_Hanoi`. In brief, the tower of Hanoi is a set of n disks with different sizes from the largest at the bottom to the smallest on the top in order. The n disks are place at one rod. The objective is to move the n stacks from one rod (`source`) to another rod (`destination`) using a rod as working space (called `via`) following some rules. (1) You can only move a disk which is on the top of the stack to the top of the stack in another place at one time. (2) A larger disk cannot be placed above a smaller disk.
First, try to consider how you can design an algorithm to handle this puzzle. It can be very interesting. The recursive algorithm is very short with only a few of lines.

```
void Hanoi(int n, char source, char destnation, char via) {
  if(n == 1)
     printf("move from %c to %c\n", source, destination);
  else {
     Hanoi(n-1, source, via, destination);
     printf("move from %c to %c\n", source, destination);
     Hanoi(n-1, via, destination, source);
  }
}
```

- Try to understand the idea of this algorithm by yourself. There are many references you can refer to in the Web. Explain the main idea of the algorithm using your own words to show your understanding.
- Show the steps to handle a set of 4 disks from source to designation by tracing the procedure.

■ **Question** 5: Consider a priority queue using the max heap. Suppose you are given a sequence of 11 numbers: 26, 5, 1, 77, 59, 61, 15, 48, 20, 80, and 8. First, insert every of the 11 numbers into the max heap in the order given as above, and show its binary tree representation after each step. Second, delete the largest number one by one from the priority queue, and show the resulting binary tree representation after each deletion.