Sample Written Qualifying Exam in CS Theory

School of Computing, National University of Singapore

1. **Algorithms**

- We consider a game where n elements numbered $1\ldots,n$ arranged in a circle. Starting from element 1, we remove every alternate element in the circle. This is done until only one element is left; the identification number of this element is called the winning position and denoted by $W(n)$.

  For example, consider 6 elements arranged in a circle as shown in Figure 1. Starting from 1, we eliminate every alternate element so we eliminate 2, 4, and 6. The situation now is shown in Figure (B). The arrow is used to denote the position of the last element that was removed. Proceeding to remove alternate elements we eliminate 3, so now get the situation shown in Figure (C). Again by eliminating the alternate element, we are left with element 5, which is the winning position. So, we have W(6) = 5.

  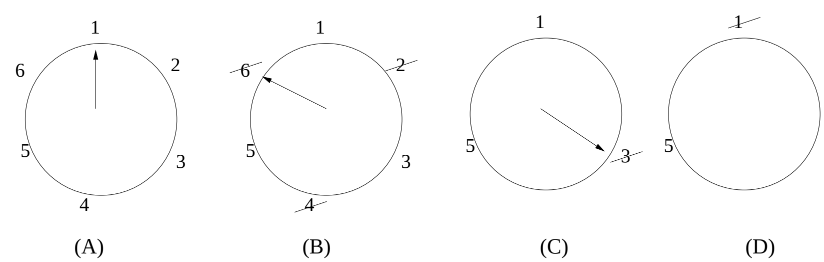

  Figure 1: The game with 6 elements

  Given a positive integer $N > 2$, how can you *efficiently* compute

  $$\sum_{2\le n\le N} W(n)$$

  i.e. the sum of the winning positions when we play the above game with 2 elements, 3 elements, ..., N elements.

- The Activity Selection Problem is defined as: given a set S of n activities $\{a_1, a_2, \ldots, a_n\}$, each activity i having a start time s_i and finish time f_i, find a maximum collection of compatible (i.e. non-overlapping) activities. Now consider the Weighted Activity Selection Problem, where each activity i also has a non-negative weight w_i, and the goal is to find a feasible solution of maximum total weight.

  (a) Characterize the optimal sub-structure for the Weighted Activity Selection Problem.

  (b) Design a Dynamic Programming algorithm to compute the optimal solution, and analyze its time complexity.

2. **Principles of Programming Languages**

- Most modern programming languages employ static scoping of identifiers. In static scoping, identifiers are declared with a specific scope in the program. Every occurrence of the identifier in that scope refers to the declaration. Assume a language where an expression of the form (`fun v -> v + 1 end` represents a function that returns its argument `x` incremented by 1. Function arguments are placed next to the called function; `g 1` applies the function to which `g` refers to the argument `1`. The scope of the variable `w` in the let-declaration `let w = 1 in` e `end` is the expression e.
  - **Static scoping**. What is the result of evaluation of the following expression under static scoping?

    ```
    let x = 3 in
       let f = fun y -> x + y end in
          let x = 7 in
              f 4+x
          end
       end
    end
    ```
  - **Dynamic scoping.** In dynamic scoping, the scope of an indentifier is dynamically extended by the function call to the body of the called function. What is the result of evaluation of the above expression under dynamic scoping?
- Consider the following pseudocode where all parameters are passed by name.

  ```
  function mystery_proc(f, i)
  begin
     integer tmp;
     tmp := 0;
     for i:= 1 to 10 do  tmp := tmp + f
     return tmp;
  end
  ```

  Assume that A is an array of integers. Explain the computation performed by invoking the above function as `mystery_proc(A[K],K)`. What value will be returned ?
- Consider the following pseudocode of class definitions using virtual functions.

  ```
  Class  A{
  Public:
  Virtual void f() { print  'a'}
  }

  Class B:A{
  Public:
         Void  f() {print 'b'}
  }

  Class C:B{
  Public:
        Void  g(){ print 'c'}
  }

  main()
  {   B x;
      C y;
      x.f(); y.f();
  }
  ```

  What are the two characters printed ? Explain your answer.

3. **Theory of Computation**

- Show that the following language is not regular.

$$L_1 = \{\sigma \in \{a,b\}^* \;\mid\; \sigma \text{ contains equal number of } a \text{ and } b\ \}$$

- Construct a context free grammar which accepts the following language.

$$L_2 = \{\sigma \in \{a,b\}^* \;\mid\; \sigma \text{ contains unequal number of } a \text{ and } b\ \}$$

4. **Logic and Formal Systems**

- Let S be the following informal statement on natural numbers

  "For any x and y, $x+y$ is even if and only if either both are even, or both are odd."

  Find a **predicate logic formula** ϕ over $\mathcal{F} = \{0, 1, +, *\}$ (function symbols: zero, one, addition, multiplication) and $\mathcal{P} = \{=\}$ (predicate symbols: equality) to describe S.

- Find a **natural deduction proof** for the following predicate logic sequent

$$\forall x(P(x) \vee (Q(x) \wedge R(x))) \vdash \forall x P(x) \vee (\exists x Q(x) \wedge \exists x R(x))$$

- Formally prove that the following program, if it terminates, correctly computes the quotient and remainder of (x/y). Does your reasoning need the assumption that division by zero is disallowed ?

```
r = x; d = 0;
while (r >= y) {
     r = r - y;  d = d + 1;
}
```