

Name: VEN THON

ID: e20191250

Group: I3-GIC-C

Assignment 13 theory information

1. What is an algorithm?

- Is a step by step procedure for calculations or for solving a problem.
- An algorithm is an effective method expressed as a finite list of well-defined instructions for calculating a function

2. What are advantages of algorithm?

- There are:
  - Used for calculation
  - data processing,
  - and automated reasoning

3. what are properties of algorithm?

- Input from a specified set
- Output from a specified set(solution)
- Definition of every step in the computation
- Correctness of output for every possible input
- Finiteness of the number of calculation step
- Effectiveness of each calculation step, and
- Generality for a class of problems

4. What are different between liner search and binary search algorithm?

- A liner search algorithm, that is, an algorithm that linearly search a sequence for a particular element.
  - Example: `liner_search(x:integer; a1,...,an: integers)`  
`i:=1`  
`while(i ≤ n and x ≠ ai)`  
`i:=i+1`  
`if i ≤ n then location :=i`  
`else location :=0`  
`<<location is the subscript of the term that equals x, or is zero if x is not found>>`
- Binary search algorithm iteratively restricts the relevant search interval until it closes in on the position of the element to be located.
  - Example: `binary_search(x:integer; a1,...,an: integers)`  
`i:=1 {i is left endpoint of search interccal}`  
`j:=n {j is left endpoint of search interccal}`  
`while(i<j)`  
`begin`  
`m:=[(i+j)/2]`  
`if x>am then i:=m+1`  
`else j:=m`  
`end`  
`if x=a1 then location :=i`  
`else location :=0`

<<location is the subscript of the term that equals x, or is zero if x is not found>>

5. Which one do you think is the best algorithm? Why?

I think a binary search algorithm is the best algorithm. Because it is related to sequence with the ordered and it have function with center element and search interval.

6. What is algorithm complexity? Give an example of it?

- In general, we are not so much interested in the time and space complexity for small inputs.
- Algorithmic complexity is concerned about how fast or slow particular algorithm performs.
- Example: \*let us assume two algorithms A and B that solve the same class of problems.

The time complexity of A is  $5,000n$ , the one for B is  $[1.1^n]$  for an input with  $n$  elements.

- For  $n=10$ , A requires 50,000 steps, but B only 3, so B seems to be superior to A.

7. What is a growth of functions? Give an example of it?

- The growth of a function is determined by the highest order term: if you add a bunch of terms, the function grows about as fast as the largest term (for large enough input values).

➤ Example:

- Show that  $f(x) = x^2 + 2x + 1$  is  $O(x^2)$ .
- For  $x > 1$  we have:

$$\begin{aligned} * x^2 + 2x + 1 &\leq x^2 + 2x^2 + x^2 \\ * \Rightarrow x^2 + 2x + 1 &\leq 4x^2 \end{aligned}$$

- Therefore, for  $C = 4$  and  $k = 1$ :  $f(x) \leq Cx^2$  whenever  $x > k$ .  $\Rightarrow f(x)$  is  $O(x^2)$