**Name - Srajan Ahuja**

**100 Number - 1001858728**

**Homework #4**

**Date - 04/20/2021**

**Q1)** Why should you be interested in learning about Lambda Calculus?

**Ans1)** One should be interested in learning about Lambda Calculus for the following reasons:-

1. Lambda Calculus can encode any computation, since it is Turing complete any programming language can be encoded using Lambda Calculus.
2. Execution models of many functional programming languages (ML, Haskell, Coq, etc.) have evolved from Lambda Calculus.
3. Languages like C#, Java, Scala are not functional but they include Lambda Calculus.

**Q2)** How do you encode the concepts of TRUE, FALSE, NOT, AND, OR?

**Ans2)**

1. TRUE = λx.λy.x
2. False = λx.λy.y
3. NOT = λa.a False True

=>NOT = (λa.a False True)True

=>NOT = (λa.a False True)(λx.λy.x)

=>NOT = (λx.λy.x) False True

=>NOT = False

1. AND = λa.λb.a b False
   1. True False

=> AND = (λa.λb.a b False) True False

=> AND = (λx.λy.x)(λx.λy.y)(λx.λy.y)

=> AND = λx.λy.y

* 1. True True

=> AND = (λa.λb.a b False) True True

=> AND = (λx.λy.x)(λx.λy.x)(λx.λy.y)

=> AND = λx.λy.x

1. OR = λa.λb.a True b

**Q3)** What is important about the Lambda Calculus expression called 'Y Combinator'?

**Ans 3)** The Y Combinator is important as it uses a combination of True and False in it’s function which allows us to use recursion in a language that doesn’t support recursion.

**Q4)** Write the Y Combinator expression in Lambda Calculus.

**Ans 4)** rec = λf.(λx.f(x x))(λx.f(x x))

**Q5)** Where did the language 'Haskell' get its name?

**Ans 5)** “Haskell” language gets its name from Haskell Curry who was one of the early logicians to work with Lambda Calculus.

**Q6)** In the video it was mentioned that Erlang was used to code what?

**Ans 6)** Erlang was used to code Whatsapp.

**Q7)** How is 'pattern matching' used?

**Ans 7)** A function can have multiple definitions corresponding to the value of the input. Pattern matching is a way to define the function on these different kinds of use cases of inputs. For example, a searching algorithm won’t perform any search if an empty array is it’s input and will return nothing, it would however perform a search on an array that has element(s), this will happen so only when the programmer will define these scenarios in the function based on the type of input (pattern matching).

**Q8)** Complete this sentence: "NP problems are hard to solve but easy to \_\_\_\_\_"

**Ans 8)** NP problems are hard to solve but easy to check.

**Q9)** What is the example of an NP problem used in the video?

**Ans 9)** Example of NP problem used is “Determining factors of a number”.

**Q10)** What are the TV shows mentioned in the video?

**Ans 10)** TV shows mentioned are -

1. The Simpsons
2. Futurama

**Q11)** Floating point numbers are essentially what?

**Ans 11)** Floating point numbers are essentially scientific notations.

**Q12)** Computers perform scientific notation in what base?

**Ans 12)** Computers perform scientific notation in base 2.

**Q13)** What is the problem with adding 1/3 + 1/3 + 1/3 using base 10 and ignoring recurring numbers?

**Ans 13)** We know that ⅓ +⅓ + ⅓ = 1. But when it comes to floating point numbers, the computer can only store up to 23 significant decimal places (32-bit) and not the entirety of recurring digits

since ⅓ = 0.33333333….

This causes ⅓ + ⅓ + ⅓ = 0.999999…… and this is known as floating point rounding error

**Q14)** What is 1/10 in base 2?

Ans 14) 1/10 in base 2 is 0.00011001

**Q15)** What is the name of the function discussed in the video?

**Ans 15)** Ackerman’s function is discussed in the video.

**Q16)** Can Ackermann's function be coded using for or 'DO' loops?

**Ans 16)** No, certain functions of a huge scale cannot be implemented using ‘for’ or ‘DO’ loops and have to implemented recursively and one of them is the Ackerman’s function.

**Q17)** What is the value of Ackermann(4,1)?

**Ans 17)** Ackerman (4,1) = 65533

**18)** How many minutes will the machine in the video take to calculate Ackermann(4,2)

**Ans 18)** The machine in the video will take 2(65533) \* 3 minutes

**Q19)** The performance characteristic of Ackermann's function is described as what?

**Ans 19)** The performance characteristic of Ackerman’s function is described as super exponential.

**Q20)** A loop nested in another loop has the performance characteristic of what?

**Ans 20)** Nested loops have characteristics of multidimensional problems.

**Q21)** What was the limitation of Fortran mentioned in the video?

**Ans 21)** Fortran didn’t allow nested loops to go more than 10 deep.

**Q22)** What real-world use needs complex recursion?

**Ans 22)** Compiler design is a real world problem which requires complex recursions in order to check complex nesting in the syntax of a code.

**Q23)** There was a need to have a language that could cope with what?

**Ans 23)** There was a need to have a language that could cope with different width of objects. A language that could perform operations on OS level efficiently unlike other High Level Languages.

**Q24)** C is most powerful when considered as the classical what?

**Ans 24)** C is most powerful when considered as the classical system implementation language.

**Q25)** What are the names of the two fields of the 'THING' structure?

**Ans 25)** Two fields of the ‘THING’ structure are -

1. char \*item - pointing towards the first character of the string .
2. struct \_thing \*next - pointing towards the next ‘THING’ type.

**Q26)** What is the advantage of the 'Triple Ref Technique'?

**Ans 26)** The ‘Triple Ref Technique’ uses a tracer to point to objects temporarily in linked lists which make it simpler to perform manipulations in Linked Lists.

**Q27)** What is the procedure used in the video to compare the different structures?

**Ans 27)** The procedure used is as follows :

1. Traversing both arrays and linked list
2. Calculating the sum of all elements visited
3. Doing the above steps 100 times
4. Calculating the average time of arrays and average time of linked lists to compare the two data structures.

**Q28)** Why is the reverse array faster on the Atari?

**Ans 28)** The reverse array is faster on Atari as on an instruction level Atari favors going backwards.

**Q29)** What would be the goal of requiring people to be exposed to coding?

**Ans 29)** People exposed to coding would understand the functioning of a computer better and in turn start thinking computationally.

**Q30)** List 3 or more of the different sort algorithms mentioned in the video

**Ans 30)**

1. Bubble sort
2. Quick sort
3. Selection sort
4. Cocktail sort
5. Heap sort

**Q31)** What is the 'Decision Problem'?

**Ans 31)** Decision Problem is a test to determine whether a certain formula/function gives a certain answer or not.

**Q32)** An example of an abstraction used in the video is, "A transistor is a type of \_\_\_"?

**Ans 32)** The example used is “A transistor is a type of switch and that switch can be opened or closed”.

**Q33)** Which video was the most interesting or your favorite?

**Ans 33)** My favorite video was Arrays vs Linked Lists - Computerphile as the reverse array being faster on Atari was a surprise to me.