

Teoría Matemática de la Computación

Primer Problemario

Prof. Miguel A. Pizaña
5 de junio de 2015

1. Dudar de todo, al menos una vez en la vida.
2. ¿Quién es Popper? ¿Qué es la falsabilidad? De un ejemplo de una afirmación falsa pero no falsable, una verdadera pero no falsable, una falsa y falsable y una verdadera y falsable.
3. ¿Qué es la duda sistemática de Descartes? ¿Qué hay que hacer, según Descartes, para alcanzar el verdadero conocimiento?
4. Explique las diferencias y similitudes entre modelo y realidad.
5. ¿Cómo hace uno para medir qué tan bien aproxima un modelo matemático a la realidad?
6. ¿Qué tan cierto es el teorema: '1+1=2'? Explique en detalle. Señale dos situaciones de la vida real en las que uno esperaría que el teorema fuera aplicable, pero de hecho no lo es. ¿Esto afecta la veracidad del teorema? ¿Cómo sabe uno cuándo puede uno aplicar el teorema y cuándo no?
7. ¿Qué tan cierto es el teorema de Pitágoras? Explique con detalle. Señale dos situaciones de la vida real en las que uno esperaría que el teorema fuera aplicable, pero de hecho no lo es. ¿Esto afecta la veracidad del teorema? ¿Cómo sabe uno cuándo puede uno aplicar el teorema y cuándo no?
8. ¿Cuáles son las ideas centrales de la Teoría de la Relatividad General?
9. ¿Qué es la longitud de Planck? ¿Cuánto mide en metros?
10. ¿Qué dice el principio holográfico? ¿Qué tiene que ver con la tridimensionalidad del universo?

11. ¿Qué dice la ley de Moore? ¿Qué tan cierta es esa ley? ¿Por cuánto más tiempo puede seguir siendo cierta esa ley? ¿Puede seguir siendo cierta indefinidamente?
12. Dentro de un millón de años, una unidad de procesamiento quiere acceder una localidad de memoria en menos de un nanosegundo. ¿Cuál es la máxima distancia del procesador a la que puede estar físicamente ubicada tal localidad de memoria? (Suponiendo que las leyes de la física conocidas no hayan cambiado demasiado).
13. Demuestre que el universo no existe.
14. Suponga que lanza usted una moneda un millón de veces y en todas esas ocasiones usted ve a la moneda caer hasta el suelo en un lindo arco parabólico. ¿Eso es garantía de que la siguiente vez que usted lance la moneda, ésta no se va a convertir en un monstruo de espagueti volador? Explique.
15. Suponga que usted juega al *melate* un millón de veces y en todas esas ocasiones usted NO se saca el premio. ¿Eso es garantía de que la siguiente vez que usted juegue *melate* tampoco se va a sacar el premio? Explique.
16. ¿Cómo explica la diferencia en sus respuestas a los dos problemas anteriores, siendo que la evidencia experimental es idéntica?
17. Pruebe que el problema del paro es indecidible.
18. Pruebe que si se agrega la restricción de que los programas corran con cierta memoria finita fija, entonces el problema del paro-con-memoria-finita, es decidible. ¿Este resultado teórico es útil en la práctica? Explique.
19. Pruebe que si C es un algoritmo de compresión sin pérdida que además satisface $|C(A)| \leq |A|$ para todo archivo A , entonces, necesariamente $|C(A)| = |A|$ para todo archivo A .
20. ¿Cuáles son los 3 principales modelos matemáticos que se usan para aproximar las nociones de “algoritmo” y “computadora”?

21. ¿Qué es una Máquina de Acceso Aleatorio Uniforme (uRAM)? ¿Qué es una Máquina de Acceso Aleatorio logarítmico (IRAM)? ¿Cómo se definen? ¿para que sirven? ¿Cuánto tiempo tardan en realizar un operación elemental? ¿Qué almacenan en cada celda de memoria? ¿cuántas celdas tienen? ¿Cuáles son las principales variaciones del modelo RAM básico? ¿Cuáles aspectos de la realidad aproximan bien estos modelos? ¿Cuáles aproxima mal? ¿Cuántos y cuáles son los supuestos falsos en los que se basan estos modelos?
22. Muestre que el algoritmo usual para multiplicar matrices tarda un tiempo del orden de n^3 en el modelo uRAM.
23. Muestre que el algoritmo usual para inicializar matrices tarda un tiempo del orden de n^2 en el modelo uRAM.
24. Muestre que el algoritmo usual para calcular el promedio de un arreglo tarda un tiempo del orden de n en el modelo uRAM.
25. Muestre que el algoritmo MergeSort tarda un tiempo del orden de $n \log(n)$ en el modelo uRAM.
26. Muestre que el algoritmo Burbuja tarda un tiempo del orden de n^2 en el modelo uRAM.
27. Si desea ordenar un arreglo de mil millones de enteros, cuanto tiempo esperaría tardarse usando: (1) Burbuja y (2) Merge sort.
28. ¿Qué consideraciones tenemos que tener a la hora de medir tiempos de ejecución de algoritmos? ¿Cuáles son las posibles fuentes de error o imprecisión al hacer mediciones?
29. El modelo uRAM predice un tiempo lineal (proporcional al tamaño del arreglo) para el algoritmo que calcula promedios de arreglos. ¿Qué tan bien se lleva esta predicción con el experimento? Explique.
30. El modelo uRAM predice que el tiempo para inicializar un arreglo por renglones es idéntico al tiempo necesarios para inicializarlo por columnas. ¿Qué se observa en el experimento? ¿Cómo explica la discrepancia?
31. ¿Qué son la jerarquía de memoria, la memoria caché, una página, un marco, el principio de localidad y la hiperpaginación? ¿De qué manera

afectan estas nociones a las predicciones del modelo uRAM y a los correspondientes experimentos de medición de tiempos de ejecución?

32. ¿Qué tan distintas pueden ser las predicciones del modelo uRAM y el modelo IRAM? Muestre un ejemplo en donde las predicciones de un modelo y otro sean muy diferentes.
33. ¿En qué condiciones es mejor usar el modelo uRAM y en qué condiciones es mejor usar el modelo IRAM?
34. Defina: “ $f(n) = O(g(n))$ ” y “ $f(n) = \Theta(g(n))$ ”. ¿Qué propiedades tienen estas relaciones? Si $f(n) = O(g(n))$ y $h(n) = O(g(n))$, ¿eso implica que $f(n) = h(n)$?
35. ¿Un 'int' es un entero? El código: `x=1;while(x++);` ¿para o se encicla? Explique en detalle.