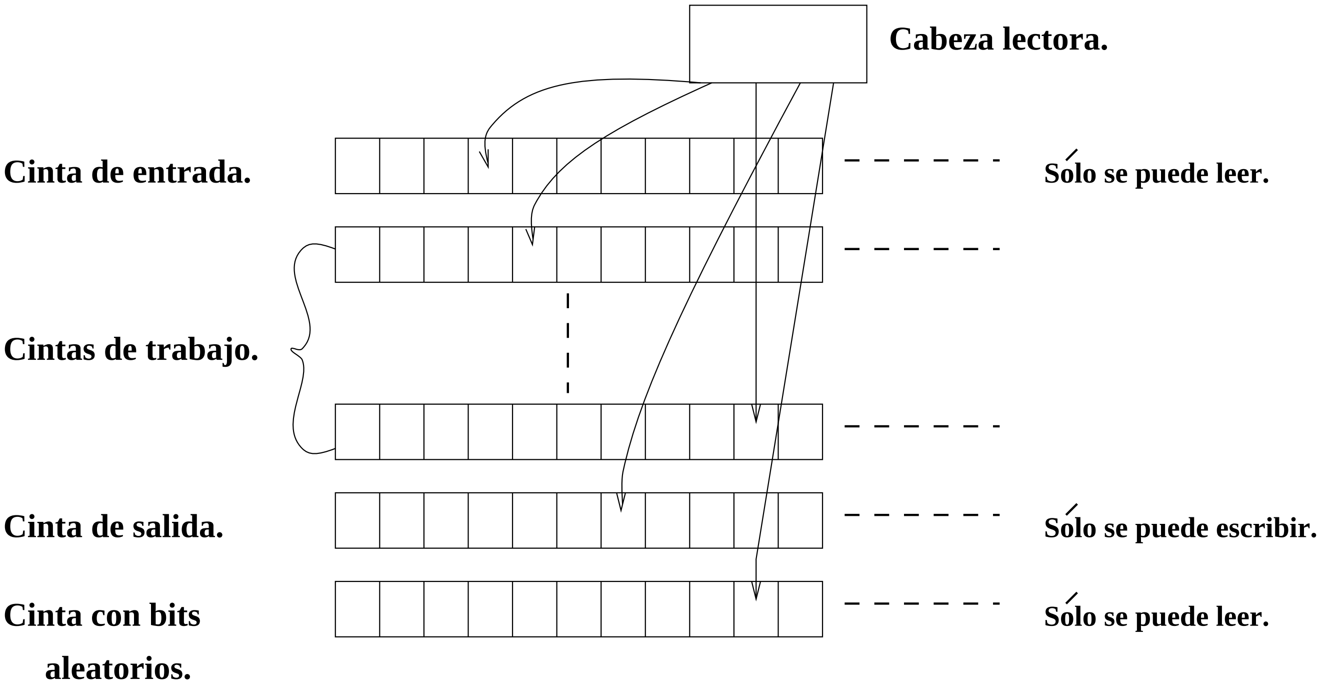
Cátedra No. 16 : 16 de Octubre

Transcriptor: Julio Aracena L.

1 Tipos de Algoritmos

La idea intuitiva de algoritmo está relacionada con el cálculo de ciertas funciones $f(x)$, donde x es la entrada. Dicho cálculo se hace mediante una máquina de Turing que recibe como entrada x. Por consiguiente, es necesario definir un algoritmo en base a un cierto tipo de máquina de Turing. Consideremos una máquina de Turing M, del siguiente tipo:



Si M se detiene en la entrada x, entonces el contenido de la cinta de salida lo denotamos $M(x)$ (indefinido si M no se detiene en x).

Ahora estamos en condiciones de definir un algoritmo y diferentes tipos de algoritmos.

Definición 1 *Un algoritmo es una función $A : \{0,1\}^* \to \{0,1\}^*$, tal que existe una máquina de Turing M, tal que $M(x) = A(x)$, cualquiera sea $x \in \{0,1\}^*$.*

Definición 2 *Diremos que un algoritmo A es polinomial si existe una máquina de Turing M, y un polinomio $p(\cdot)$ tal que*

i) $M(x) = A(x), \forall x \in \{0,1\}^*$, *y*

ii) El número de pasos que ejecuta M antes de detenerse es menor o igul a $p(n)$, en entradas de largo n

Algunos tipos de algoritmos son:

- **Monte Carlo**: un algoritmo A es Monte Carlo si existe una máquina de Turing probabilista M, y un polinomio $p(\cdot)$ tal que :
  - i) $P[M(x) = A(x)] \geq \frac{2}{3}$, $\forall x \in \{0,1\}^*$
  - ii) El número de pasos que ejecuta M antes de detenerse es menor o igual a $p(n)$, en entradas de largo n.
- **Las Vegas**: un algoritmo A es Las Vegas si existe una máquina de Turing probabilista M, y un polinomio $p(\cdot)$ tal que:
  - i) $M(x) = A(x), \forall x \in \{0,1\}^*$.
  - ii) El tiempo esperado que M toma en detenerse en entradas de largo n es a lo más $p(n)$.
- **Polinomiales No-Uniformes**: Si existe una secuencia infinita de máquinas de Turing: $M_1, M_2, \ldots$, y dos polinomios $p(\cdot)$ y $q(\cdot)$, tal que:
  - i) $M_n(x) = A(x), \forall x \in \{0,1\}^n$.
  - ii) M_n se detiene en a lo más $p(n)$ pasos en entradas de tamaño n.
  - iii) La descripción de M_n (en alguna codificación standard de las máquinas de Turing) está acotada por $q(n)$.

De esta forma, la clase RP está formada por aquellos lenguajes que son decidibles por un algoritmo tipo Monte Carlo, y la clase ZPP está formada por los lenguajes de RP $\cap$ coRP con algoritmo tipo Las Vegas que los deciden. [1]
Un criptosistema pude ser seguro para un cierto tipo de adversario y para otro no, dependiendo de la capacidad de cada adversario. Es necesario definir, por lo tanto, los tipos de adversarios que existen.

[1] *Computational Complexity*, C.H. Papadimitriou, Addison Wesley,1994.

2 Tipos de Adversarios

Un adversario es un algoritmo eficiente (i.e., probabilista o polinomial), o un algoritmo polinomial no-uniforme. En el primer caso hablamos simplemente de adversario, y en el segundo de adversario no-uniforme. En este último caso un adversario puede usar diferentes algoritmos para diferentes tamaños de las entradas.
Es posible demostrar que un adversario no-uniforme, es más fuerte que un simple adversario. Luego, probar que un criptosistema es seguro en presencia de un adversario no-uniforme es mejor que probarlo frente sólo a un adversario.

3 Funciones de un sentido

La más débil hipótesis que se puede hacer en criptografía frente a un adversario, es que $P \neq NP$. Sin embargo, esto no es suficiente, ya que suponiendo que existe $L \in NP, L \notin P$, entonces tal vez, $L \in BPP$, pues es un problema abierto que $BPP \subseteq NP$, pero nuestro adversario tiene acceso a algoritmos probabilísticos.
Tampoco es suficiente suponer que : $BPP \neq NP$, pues esto sólo significa que si existe $L \in NP, L \notin BPP$, entonces todos los adversario deciden incorrectamente, con una probabilidad mayor que un tercio, sobre un número infinito de entradas x, si $x \in L$. Estas decisiones erróneas, aunque en un número infinito, pueden ocurrir muy infrecuentemente, por ejemplo, una por cada tamaño de la entrada x, permitiendo de esta forma quebrar nuestro criptosistema en forma frecuente. Esto último es contrario a la idea de un criptosistema seguro.
De esta forma, necesitamos una hipótesis más fuerte que nos garantice que todo adversario, para toda entrada de tamaño suficientemente grande n, tome una decision equivocada con alta probabilidad de decidir si $x \in L$, para algún $L \in NP$, es decir, que se equivoque frecuentemente sobre todas las entradas de tamaño n. Y por otro lado, dado n suficientemente grande, poder generar una entrada x de tamaño n tal que todo adversario se equivoque en la decisión sobre x.
La hipótesis que garantiza lo anterior es la existencia de funciones de un sentido y de puerta secreta. A continuación definimos la funciones de un sentido.

Definición 3 *Una función $f : \{0,1\}^* \to \{0,1\}^*$ se dice de un sentido fuerte si :*

i) Existe M, máquina de Turing determinista polinomial tal que $M(x) = f(x)$, cualquiera sea $x \in \{0,1\}0^*$.

ii) Para todo polinomio p y para todo adversario A, $\exists n_0 \in \mathbb{N}, \forall n \geq n_0$

$$Pr[A(f(x)) \in f^{-1}(f(x))] \leq \frac{1}{p(n)},$$

donde la probabilidad es sobre los x en $\{0,1\}^n$ *y la fuente de aleatoridad de A.*

Es decir, las funciones de un sentido fuerte son aquellas que son invertibles en una proporción pequeña (a lo más $\frac{1}{q(n)}$) de ellas, en entradas de tamaño n.
Dado que $f(x)$ es calculable en un tiempo polinomial, $\exists c, \forall x\,; |f(x)| < |x|^c$. Por lo tanto, nuestro interés está basado en funciones tales que $\exists k,\ |f(x)| \geq |x|^k$. De lo contrario, debiera ser fácil de invertir f.
Consideremos, por ejemplo, la función: $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ con $f(x,y) = x \cdot y$. Esta función puede ser fácilmente invertible en salidas de tipo par, que representan la mitad de sus salidas. Luego, no es una función de un sentido fuerte. Sin embargo, f es difícil de invertir en salidas que son productos de dos números primos de tamaño similares, que corresponde a una fracción de tamaño polinomial sobre el conjunto de enteros de largo n. Esto motiva la siguiente definición:

Definición 4 *Una función* $f : \{0,1\}^* \to \{0,1\}^*$ *se dice de un sentido débil si:*

i) Existe M, máquina de Turing determinista polinomial tal que $M(x) = f(x)$, *cualquiera sea* $x \in \{0,1\}0^*$.

ii) Existe un polinomio $q(\cdot)$ *y para todo adversario A,* $\exists n_0 \in \mathbb{N}, \forall n \geq n_0$

$$Pr[A(f(x)) \notin f^{-1}(f(x))] \geq \frac{1}{q(n)},$$

donde la probabilidad es sobre los $x \in \{0,1\}^n$ *y la fuente de aleatoridad de A.*

Es decir, las funciones de un sentido débil, son aquellas que no son invertibles en una proporción de $\frac{1}{q(n)}$ sobre el conjunto de entradas de tamaño n.
Si consideramos, de nuevo la funcion $f(x,y) = x \cdot y$, si $|x| = n$, entonces $P[x$ es primo$]$ $\approx \frac{1}{n}$, luego $P[x, y$ son primos$] \approx \frac{1}{n^2}$, si $|x| = |y| = n$. Así, para todo n, sobre $1 - \frac{1}{n^2}$ de los pares de entradas de f de largo $2n$ son primos de igual largo. Si aceptamos que ningún adversario puede invertir f cuando x e y son primos de un mismo largo, entonces f es una función de un sentido débil, con $q(n) = O(n^2)$.