

Práctico ALGABO: Algoritmos probabilísticos y algoritmos genéticos.

Mauricio Velasco

1. (*Estadística en el problema de selección de secretarie*) Implemente el Algoritmo de generación de permutaciones uniforme mediante sorting en S_n que vimos en clase.
 - a) Escriba el código de su implementación
 - b) Para $n = 20$ candidatos genere una muestra de $m = 100$ permutaciones y escriba las 5 primeras y las 5 últimas permutaciones de su muestra.
 - c) En las permutaciones obtenidas en el punto anterior, marque los candidatos que la firma debería contratar, si el ranqueo de candidatos esta dado por tales permutaciones (en el problema de selección de asistente que vimos en clase).
 - d) Escriba el código de un programa que reciba una permutación y calcule el número de candidatos que la firma debería contratar en el problema de selección de asistente que vimos en clase).
 - e) Para muestras de $m = 100$, $m = 500$ y $m = 1000$ permutaciones haga un dibujo que contenga, en un solo par de ejes:
 - 1) Un histograma del número de contrataciones realizadas en cada permutación de su muestra (ver por ejemplo https://www.w3schools.com/python/matplotlib_histograms.asp)
 - 2) Una recta vertical en el valor teórico que vimos en clase.
 - 3) Una recta vertical en el número promedio de contrataciones realizadas.
2. (*Puntos fijos en permutaciones aleatorias*) Recuerde que los puntos fijos de una permutación $\sigma \in S_n$ son aquellos índices $i \in \{1, \dots, n\}$ con $\sigma(i) = i$.

- a) Para un entero j cualquiera defina la variable aleatoria $Y^{(j)} : S_n \rightarrow$ dada por

$$Y^{(j)}(\sigma) = \begin{cases} 1, & \text{si } \sigma(j) = j \\ 0, & \text{de lo contrario} \end{cases}$$

Si \mathbb{P} es la medida de probabilidad uniforme en S_n , calcule $[Y^{(j)}]$ dando un argumento preciso para su respuesta.

- b) Use la parte (a) para calcular el número esperado de puntos fijos de una permutación aleatoria de S_n , elegida uniformemente.

3. (*Un algoritmo genético innecesario*) Sea X el conjunto de cadenas binarias de longitud 5. Cada cadena de estas es la representación binaria de un número entero entre 0 y 32 inclusive. Sea $f(x) = x^3$. Usaremos algoritmos genéticos para maximizar f en X .

- a) Escriba el código de un algoritmo genético que intente maximizar la función $f(x)$ para $x \in X$. Su código debe incluir funciones de **reproducción**, de **cross-over** y de **mutación** donde la última depende de una probabilidad de mutación q a ser especificada.
- b) Construya una población inicial aleatoria I_0 de 4 cadenas. Ejecute su algoritmo un total de 10 pasos y produzca una sola imagen con las gráficas de los valores mínimo, promedio y máximo de la función objetivo en su población (el eje x de su dibujo es el tiempo y el eje y las unidades de la función objetivo y debe contener tres curvas).
- c) Haga el dibujo de la parte (b) quitándole a su algoritmo la fase de **cross-over**. Como cambian sus resultados?