

Teoría de Algoritmos I (75.29 / 95.06) Ejercicios de Algoritmos randomizados

- 1) Un nuevo regente del país “Atrasoñia” llegó al poder de la mano de 3 aliados. Como todo monarca decide repartir los puestos clave de poder para dejar contentos a todos ellos. Por otro lado, no desea que se formen cúmulos de poder que dificulten su dominio. Para ayudarse ha generado un diagrama donde se pueden ver los diferentes puestos y con qué otros puestos claves se relaciona. Lo que intenta es lograr que para cada par de puestos relacionados entre sí, las facciones que lo controlen sean diferentes. Existe un número total de relaciones entre puestos y lo que desea es que la mayor cantidad posible de ellos cumplan el criterio de “balanceo de poder”

a) Proponer un algoritmo randomizado para lograr al menos $\frac{2}{3}$ del total posible de relaciones balanceados.

- 2) El problema de set independiente corresponde a la clase NP-C. No obstante, dado la gran cantidad de aplicaciones de este, nos interesaría contar con algún mecanismo para obtener dado un grafo $G=(V,E)$ el subconjunto de vértices independientes de mayor tamaño posible.

Acotamos el problema a un grafo con n vértices, donde cada uno de ellos tiene exactamente d ejes. Analizar el siguiente algoritmo randomizado:

Para cada vértice i seleccionar de forma uniformemente aleatoria x_i el valor 0 o 1. Seleccionar para el set independiente únicamente aquellos vértices con el valor 1 que no se conecten mediante ejes con otros vértices que tengan el valor 1.

- a) Probar que el set resultante es independiente.
- b) Calcular el tamaño esperado del set independiente (en función de n y d)
- c) Para el cálculo de x_i utilizamos $p=\frac{1}{2}$. Determinar si este es el valor más conveniente o podemos encontrar uno mejor. Si es así, de la fórmula del valor esperado con esta nueva probabilidad.

- 3) Un estudio de videojuegos está lanzando un nuevo juego para celular que permite competir entre 2 jugadores. Cada jugador tiene un conjunto de criaturas de las que debe elegir un subconjunto de tamaño N . Una vez que se elijan se enfrentarán uno a uno. Para lograr una pelea más emocionante se mezclan los conjuntos. De esa forma en cada batalla van apareciendo de forma aleatoria las criaturas elegidas.

El siguiente es el pseudocódigo que han implementado:

```

Sea mezclado un vector de 1 a n vacío
Sea criatura un vector de 1 a n con las criaturas
seleccionadas
Desde i=1 a n
    x = random(1..n)
    Sea encontrado = false
    While not encontrado:
        if (criatura[x] != null):
            mezclado[i] = criatura[x]
            criatura[x] = null
            encontrado = true
        else:
            x++
    if (x > n):
        x = random(1..n)

```

Determinar si es un buen método de mezclado aleatorio. En caso negativo proponer otro que lo sea.