CC40A - Guía 3.5

1 Algoritmos Aproximados

1. Dé un ejemplo donde la aproximación del vertex cover siempre entrega el doble de vértices que lo óptimo.

2. Diseñe un algoritmo lineal para el problema del vertex-cover cuando el grafo es en realidad un árbol (grafo acíclico).

3. Demuestre que en un circuito óptimo sobre un grafo de ciudades en el plano, donde el costo es la distancia euclidiana, nunca se cruzan los caminos.

2 Algoritmos Probabilísticos

1. Suponga que tiene un generador aleatorio que le entrega un valor entre 1 y k, para cualquier k que le pida. Muestre cómo lo usaría para generar permutaciones aleatorias eficientemente, de modo que todas las permutaciones tengan la misma probabilidad de ser generadas.

2. Suponga que tiene una lista encadenada ordenada pero implementada en arreglos. Es decir, tiene $L[1..n]$ y ademas $Next[1..n]$ y $Prim$. El menor elemento es $L[Prim]$, el segundo es $L[Next[Prim]]$, el tercero es $L[Next[Next[Prim]]]$ y así. Muestre primero que un algoritmo determinístico de búsqueda es $\Omega(n)$ en promedio. Luego, muestre un algoritmo probabilístico que sea $O(\sqrt{n})$ en promedio. Ayuda: qué pasa si prueba en k posiciones aleatorias del arreglo? Cuál es el costo promedio? Cuál es el k óptimo?

3. Considere un algoritmo Monte Carlo para un problema, cuyo costo de ejecución es $T(n)$ en el peor caso, y que produce una respuesta correcta con probabilidad $p(n)$. Suponga que dada una solución, el costo de verificar si es correcta o no es $C(n)$. Muestre que existe un algoritmo Las Vegas (o sea, nunca se equivoca) cuyo costo promedio es $O((T(n) + C(n))/p(n))$.

4. Sea un algoritmo Monte Carlo de complejidad $T(n)$ que erra con probabilidad p. Queremos convertirlo en otro que erra con probabilidad $q < p$. Muestre cómo lo haría y qué complejidad tiene el nuevo algoritmo.