CC40A: Diseño y Análisis de Algoritmos
Pauta del Examen - Semestre Otoño 2001

Profesor: Ricardo Baeza Yates, Prof. Auxiliar: Felipe Saint-Jean

Tiempo: 2:30 hras, con apuntes y en hojas separadas

P1 (2.0 ptos.)

a) *Suponga que tiene dos arreglos S_1 y S_2 con n números reales cada uno y un valor entero x. Determinar si existen dos elementos, uno de cada arreglo, tal que el valor absoluto de su resta es x si los arreglos están desordenados. Trate de obtener la mejor complejidad temporal posible.*

Como debemos buscar muchos pares, es mejor ordenar ambos arreglos usando Merge o Heap sort para tener $O(n \log n)$ en el peor caso. Luego, para cada elemento y de S_1 buscamos usando búsqueda binaria en S_2, si está $y - x$ o $y + x$. Apenas encontremos un par, lo devolvemos. Como en el peor caso hacemos n búsquedas binarias, la complejidad total es $O(n \log n)$.

b) *Suponga ahora que tiene dos arreglos ordenados de tamaño m y n ($m \leq n$) y queremos calcular la intersección de ambos conjuntos. Diseñe un algoritmo que sea eficiente tanto si m es pequeño con respecto a n como si es parecido. Indique la complejidad de su algoritmo.*

Si m es pequeño lo más rápido es hacer m búsquedas binarias en el arreglo grande. Sino, es mejor hacer una mezcla (*merge*) aprovechando que están ordenados. La complejidad del primer algoritmo es $m \log n$ y el segundo es $m + n$, así que el punto de corte es cuando $m = n/(\log n - 1)$. La complejidad final es $O(\min(m \log n, m + n))$.

P2 (2.0 ptos.)

El juego del Black Jack o Veintiuna Real consiste en llegar lo más cerca a 21 sin pasarse, tal que las figuras valen 10 puntos, los ases pueden valer once o un punto y el resto de las cartas su valor nominal. El casino reparte dos cartas y luego pregunta a todos los jugadores (incluyendo el mismo) si uno quiere más cartas hasta que cada jugador dice que no quiere seguir o tiene más de 21. Hay una apuesta inicial de i pesos y luego se pueden apostar i pesos más. El casino paga sólo si uno consigue más puntos que el (pero no pasando los 21).

a) *Diseñe un algoritmo en línea para jugar este juego y calcule la ganancia promedio cuando decide usar 2 cartas, suponiendo que todas las cartas posibles tienen la misma probabilidad y que el casino siempre se reparte sólo dos cartas.*

Un algoritmo podría ser: si tengo 20 o 21 doblo la apuesta. Si tengo 18 o 19 me quedo. Si tengo 15 o 16 me retiro la primera vez y si no me quedo. Si tengo menos pido otra carta. Luego de calcular la nueva suma, aplico nuevamente los mismos pasos. Si me paso, no lo digo por si se pasa el casino (0.7 puntos).

En el algoritmo anterior sólo decido usar 2 cartas si tengo 18 o más (porque si me retiro no pierdo no gano nada). Luego la ganancia será i o $2i$ menos o más dependiendo si gano o no en alguno de esos dos casos. El número de pares posibles de cartas es $13^2 = 169$. El número de pares que dan 20 o 21 es 24 y los que dan 18 o 19 son 17. Luego, la probabilidad de ganar el doble es $1-24/169$ y la probabilidad de ganar la apuesta inicial es $1-41/169$. Ponderando esto por los dos posibles casos ($24/41$ o $17/41$), obtenemos una ganancia aproximada de $1.05i$. Lo que quiere decir que cuando decido usar 2 cartas arriesgo muy poco, pues en promedio gano (0.3 puntos).

b) *¿Cómo definiría competitividad en este juego? Calcule la competitividad para el caso anterior.*

El caso óptimo aquí es cuando se que cartas tiene el casino y entonces se cuando debo retirarme o doblar la apuesta. Como la ganancia aumenta en el caso óptimo, usamos el inverso. Es decir, la competividad sería $Ganancia(optimo)/Ganancia(algoritmo)$, considerando la ganancia promedio ya que es un juego de azar (0.5 puntos). El peor caso siempre es perder, así que no tiene sentido usarlo.

En el caso anterior nunca pierdo y gano cuando tengo más que el casino. Rehaciendo los cálculos obtengo aproximadamente $1.3i$ lo que da una competitividad de alrededor de 1.25 (0.2 puntos).

c) *¿Serviría agregar decisiones aleatorias a su algoritmo? ¿Qué haría?*

Puede servir, en particular en los casos en que tengo duda y a veces podría correr el riesgo (por ejemplo, decidir aleatoriamente no retirarse si tengo 15 o 16 y luego aleatoriamente decidir pedir otra carta o no en ese caso). Además ya que es un juego de azar, el adversario sabe menos si uso aleatoridad (0.3 puntos).

P3 (2.0 ptos.)

Suponga que Ud. quiere resolver el problema de colorear un grafo con el número mínimo de colores de modo que no queden vértices adyacentes del mismo color usando un algoritmo genético.

a) *¿Cómo codificaría cada instancia del problema? (genes o cromosomas)*

Hay muchas maneras, por ejemplo podrían ser kn bits si hay n vértices y podemos codificar los colores en k bits.

b) *¿Que función de calidad usaría? (fitness)*

Tiene que ser una función del número de colores, mientras más pequeño mejor.

c) *¿Que operaciones usaría y cómo las adaptaría en este caso?*

Se pueden usar todas las operaciones, pero hay que usarlas de modo de mantener la integridad de los vértices en los cromosomas y de eliminar secuencias donde dos vértices adyacentes tienen el mismo color (esas morirían).