UNIVERSIDAD DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

Matemáticas Discretas para la Computación - CC3101
Control 4 - Semestre Primavera 2013

P1 (2.0 pt)

Los profesores Moe, Larry y Curly han diseñado un nuevo método de ordenamiento, el cual en este control vamos a dar por correcto. Este se invoca como `StoogeSort (`A`,`1`,`n`)`:

```
StoogeSort(int[] A, int i, int j)
{
  if (A[i] > A[j])
      swap (A[i], A[j]);
  if (i + 1 >= j) return;
  k = floor((j - i + 1)/3);
  StoogeSort(A, i, j - k); /* Primeros dos tercios */
  StoogeSort(A, i + k, j); /* Ultimos dos tercios  */
  StoogeSort(A, i, j - k); /* Otra vez los primeros dos tercios */
}
```

1. Calcule la complejidad del algoritmo y determine si es mejor, igual o peor que el $O(n \log n)$ que consiguen varios algoritmos conocidos.

2. Obtenga una versión del Teorema Maestro para una recurrencia del tipo $T(1) = b, T(n) = aT(n/c)+ b$, con $c > 1$.

Solución:

1. $T(1) = 1$ y $T(n) = 1 + 3T((2/3)n)$, sea $n = (3/2)^k$ y $H(k) = T(n)$, entonces $H(0) = 1$ y $H(k) = 1+3H(k-1)$. Entonces $H(k+2)-H(k+1) = H(k+1)-3H(k)$, o $H(k+2)-4H(k+1)+3H(k) = 0$. El polinomio característico es $x^2-4x+3 = 0$ con raíces $x = 3$ y $x = 1$. Entonces $H(k) = a\,3^k + b$, con $H(0) = 1 = a + b$ y $H(1) = 1 + 3H(0) = 4 = 3a + b$, de lo que se deduce $a = 3/2$ y $b = -1/2$, y entonces $H(k) = 3^{k+1}/2-1/2$. Finalmente, $T(n) = (3/2)3^{\log_{3/2} n}-1/2 = (3/2)n^{\log_{3/2} 3}-1/2 = O(n^{\log_{3/2} 3})$. Se ve que el exponente es mayor que 1 (es $O(n^{2,71})$), por lo que es peor que $O(n \log n)$. También se puede hacer como caso particular del que sigue.

2. Definiendo $n = c^k$ y $H(k) = T(n)$, tenemos $H(0) = b$ y $H(k) = aH(k-1)+b$. Definiendo $Y(k) = H(k)/a^k$, tenemos $Y(0) = b$ y $Y(k) = Y(k-1) + b/a^k$, así $Y(k) = b/a^k + b/a^{k-1} + \ldots + b/a^1 + b/a^0 = b\sum_{i=0}^{k} 1/a^i$. Si $a = 1$ esto es $Y(k) = b(k+1)$, $H(k) = b(k+1)$ y $T(n) = b(1+\log_c n) =$

$O(\log n)$. Si $a > 1$, esto es $Y(k) = b\,(a - 1/a^k)/(a-1)$, entonces $H(k) = b(a^{k+1} - 1)/(a-1)$ y $T(n) = O(n^{\log_c a})$, lo cual resulta $O(n)$ si $a = c$, superlineal si $a > c$, y sublineal si $a < c$. Finalmente, si $a < 1$, la suma es similar, $Y(k) = b(1/a^k - a)/(1-a)$, $H(k) = b(1 - a^{k+1})/(1-a)$ y $T(n) = b(1 - an^{\log_c a})/(1-a) = O(1)$ pues $a < 1 < c$. En el caso de arriba tenemos $a = 3$ y $c = 3/2$, por lo que el costo es $O(n^{\log_c a}) = O(n^{\log_{3/2} 3})$.

P2 (2.0 pt)

Un *poliedro regular* es un polígono convexo formado por un cierto número f de caras, todas iguales a un polígono regular de k aristas del mismo largo y k vértices. Por ejemplo un tetraedro tiene $f = 4$ caras, cada una de ellas un triángulo equilátero ($k = 3$). Un cubo tiene $f = 6$ caras, cada una de ellas un cuadrado ($k = 4$). Si consideramos los vértices y aristas del poliedro como un grafo, es sabido que este grafo es planar, con f regiones.

1. Considere el caso en que los vértices del poliedro tienen todos grado 3. Use la fórmula de Euler para encontrar cuáles son todos los poliedros posibles en este caso, su número total de vértices, aristas y caras.
2. Repita la parte 1 considerando grados 4 y 5.
3. Muestre que los poliedros regulares obtenidos son todos los posibles.

Solución: Cada una de las f caras tiene k lados (aristas), y cada arista sirve para dos caras, por lo que se cumple $kf = 2e$; esta idea ya se vio en una clase de planaridad. Si suponemos grado a para todos los nodos, vale $av = 2e$ por el teorema de los saludos. Aplicando la fórmula de Euler $v - e + f = 2$ tenemos $(2/a)e - e + (2/k)e = 2$, es decir $e = 2/(2/a - 1 + 2/k)$.

1. Con $a = 3$ tenemos $e = 2/(2/k - 1/3)$, lo que tiene solución para $k = 3$ ($e = 6$, $v = 4$, $f = 4$, tetraedro), $k = 4$ ($e = 12$, $v = 8$, $f = 6$, cubo), y $k = 5$ ($e = 30$, $v = 20$, $f = 12$, dodecaedro).
2. Para grado $a = 4$, tenemos la fórmula $e = 2/(2/a - 1 + 2/k)$. Entonces hay solución para $k = 3$ ($e = 12$, $v = 6$, $f = 8$, octaedro) y nada más. Para grado $a = 5$ tenemos solución para $k = 3$ ($e = 30$, $v = 12$, $f = 20$, icosaedro) y nada más.
3. No hay más opciones porque en todo grafo planar existe al menos un vértice de grado 5 o menos (teorema visto en clase).

P3 (2.0 pt)

En un grafo simple se han asignado costos positivos a las aristas, $c : E \to \mathbb{R}^+$. Luego se ha elegido un nodo v, y se ha calculado un camino de costo mínimo entre v y cada $u \in V - \{v\}$. Muestre que la unión de todos esos caminos es un árbol cobertor (spanning tree), o que si no lo es se puede convertir en uno sin aumentar el costo de ningún camino.

Obs: la unión de esos caminos *no* es necesariamente lo mismo que se obtiene si se recuerdan las aristas involucradas en el algoritmo de Dijkstra. Use sólo la propiedad de que es la unión de caminos mínimos, sin suponer algún algoritmo en particular.

Solución:

Forman un árbol a menos que hayan dos caminos a nodos u y u' que se crucen en un nodo w (notar que u o u' podrían ser el mismo w). Pero ambos caminos a w deben tener el mismo costo (óptimo) pues sino podríamos reducir el costo del camino a u o a u' eligiendo llegar a w por el otro camino, lo que sería contradictorio con la hipótesis de que los caminos eran óptimos. Por lo tanto podemos eliminar una de las dos aristas que llegan a w, y se puede llegar tanto a u como a u' por el otro camino al mismo costo. Las demás aristas se pueden eliminar con el mismo razonamiento, si es que no son las únicas que llegan a los nodos que preceden a w.