Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática

Algoritmos y Estructuras de Datos

Guía de Trabajos Prácticos número 1
Diseño y análisis de algoritmos

[1] [Teoría y Operativos]

a) Ordenar las siguientes funciones en orden creciente de tiempo de ejecución. Además, para cada una de las funciones $T_1, \ldots, T_5$ determinar su velocidad de crecimiento (expresarlo con la notación $O(\,\cdot\,)$).

$$\begin{aligned}
T_1 &= n^2 + 2 \cdot 4^n + 5^3 \\
T_2 &= 3 \cdot 2^n + 5n^4 + 2\,n \\
T_3 &= 2 \log_5 n + \sqrt{n} + 3\,n^2 + 2\,n^5 \\
T_4 &= 4^5 + 2{,}7 \log_4 n + \log_2(5) n^{1{,}5} \\
T_5 &= \log_2 n + 5
\end{aligned}$$

b) ¿Porqué decimos que $(n+1)^2 = O(n^2)$ si siempre es $(n+1)^2 > n^2$?

c) Suponga que cierto procesador tiene una frecuencia de reloj de 3 Ghz, y simplifique el análisis suponiendo que por cada tres ciclos de reloj se realiza una operación en particular. Estime de qué tamaño puede ser el problema que se puede resolver en un segundo usando un algoritmo que requiere $T(n)$ operaciones, con los siguientes valores de $T(n)$: $\log(n)$, n, $n\log(n)$, n^2, 2^n y $n!$.

d) Una manera útil de pensar acerca del crecimiento de complejidad computacional es considerar como varía el tiempo de cómputo si el tamaño del problema se duplica. Determine el incremento de costo para: $T(n)$: 1, $\log(n)$, n, $n\log(n)$, n^2, n^3 y 2^n.

e) Dé ejemplos de algoritmos cuyo órden de complejidad algorítmica $O(n)$ sea: $O_1(n) = 1$, $O_2(n) = \log(n)$, $O_3(n) = n$, $O_4(n) = n\log(n)$, $O_5(n) = n^2$, $O_6(n) = 2^n$, $O_7(n) = n!$.

f) Considere el siguiente algoritmo:

```
bool algoritmo(list<int> &L, list<int>::iterator p, int x) {
       if(p==L.end()) return false;
       if(*p==x) return true;
       return algoritmo(L,++p,x);
    }
```

- Comente qué se implementa.
- Determine el tiempo de ejecución $T(n)$ para:
  1) El peor caso $T_{\text{peor}}(n)$
  2) El mejor caso $T_{\text{mejor}}(n)$
  3) El caso promedio $T_{\text{prom}}(n)$
- Determine el órden de complejidad algorítmica $O(n)$.

g) Dado el siguiente par de funciones:

```
void f(int n, vector<bool>& VL){
    if(n==0){
        for(auto x:VL) cout<<x<<" ";
        cout<<endl;
    }else{
        auto VL_0 = VL;
        auto VL_1 = VL;
        VL_0.push_back(0);
        VL_1.push_back(1);
        f(n-1,VL_0);
```

```
        f(n-1,VL_1);
    }
}
void f(int n){
    vector<bool> VL;
    f(n,VL);
}
```

- Renombrar la función con un nombre alusivo a la tarea que efectúa. Justifique.
- Determinar el orden algorítmico de `f` respecto al parámetro `n`

h) Explique que quiere decir la propiedad de "Transitividad" de $O()$.

i) Explique la "Regla del producto" para $O()$.

j) Explique cuándo dos tasas de crecimiento del orden del tiempo de ejecución son equivalentes si.

k) Utilizando la estrategia de coloreado de grafos modele los siguientes problemas:

1) *Fixture*. Participan seis equipos en un campeonato: Boca, Colón, Estudiantes, Talleres, Unión y Vélez. Vélez ya jugó con Boca y Unión; Colón ya jugó con Boca y Unión; y Talleres jugó con Estudiantes y Unión. Cada equipo juega un partido por semana. Elabore un calendario de forma que tal que cada equipo juegue una vez con todos los demás en el menor número posible de semanas. Sugerencia: cree un grafo cuyos vértices representen pares de equipos que aún no han jugado entre sí. ¿Cómo deberían ser las aristas para que en una coloración lícita del grafo, cada color pueda representar los partidos que se jugaron en una semana?
2) *Programación de exámenes finales*. Se requiere programar los exámenes finales de una universidad de manera de que ningún estudiante tenga dos exámenes al mismo tiempo.
3) *Asignación de frecuencias*. Se requiere asignar canales de frecuencias a estaciones emisoras de televisión por aire de tal manera de que no haya dos estaciones en un radio de 250 kilómetros que operen en el mismo canal.

l) Queremos multiplicar cuatro matrices $M_1 \times M_2 \times M_3 \times M_4$ de números reales, donde M_1 de 10 filas por 20 columnas, M_2 de 20 por 50, M_3 de 50 por 1 y M_4 de 1 por 100. Asuma que la multiplicación de una matriz $p \times q$ por una $q \times r$ requiere pqr operaciones escalares (número de operaciones requerido por el algoritmo común de multiplicación de matrices). Encuentre el orden óptimo en que se deben multiplicar las matrices para minimizar el número total de operaciones escalares. Cómo se podría encontrar el orden óptimo si hay un cantidad arbitraria de matrices de dimensión arbitraria ?

m) Considere un conjunto de n ciudades y una tabla de distancias entre pares de ellas. Escriba un pseudocódigo para encontrar un camino corto que pasa una vez por cada ciudad y vuelva a la ciudad de la cual partió. No se conoce un método para obtener el camino más corto excepto a través de una búsqueda exhaustiva. En consecuencia, trate de encontrar un algoritmo eficiente para este problema usando reglas heurísticas razonables.