



Instituto Politécnico Nacional

Escuela Superior de Cómputo



## Sesión 3: Más algoritmos

Integrantes:

Escutia López Arturo

López Santiago Daniel

12/Sept/2016

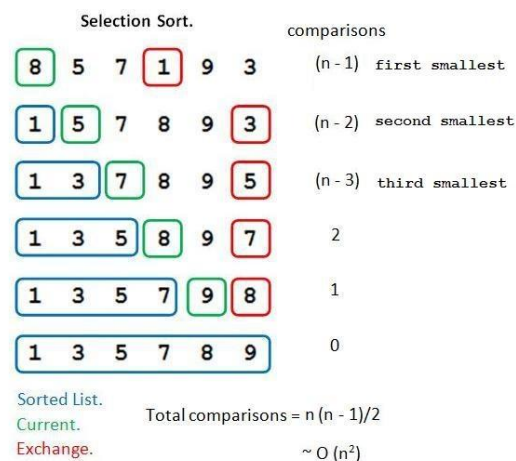
```

#include <bits/stdc++.h>
using namespace std;

void selection(vector<int> nums){
    int min,aux,j,i;
    ofstream myfile;
    for(i=0;i<nums.size();i++){
        min=i;
        for(j=i+1;j<nums.size();j++){
            if(nums[j]<nums[min])
                min=j;
        }
        if(min!=i){
            aux=nums[i];
            nums[i]=nums[min];
            nums[min]=aux;
        }
    }
    i=0;
    myfile.open ("ordered.txt");
    while(i<nums.size()){
        myfile<<nums[i]<< "\n";
        i=i+1;
    }
    myfile.close();
}

```

- 1.1 En este programa hacemos uso del algoritmo de ordenamiento selection sort. La idea es simple. Busco el elemento más pequeño de la lista, y lo intercambio con el elemento de la primera posición... busco el segundo elemento más pequeño y lo intercambio con el elemento de la segunda posición. Con esto se hace más o menos el mismo número de comparaciones que se hacen con el algoritmo de burbuja tradicional, pero con muchos menos movimientos de memoria, y por lo tanto con un rendimiento claramente superior, de esta forma luciría el ordenamiento de un arreglo.



```

#include <bits/stdc++.h>
using namespace std;
typedef long long int64;
int64 mult(int64,int64);
int main(int argc, char * argv[]) {
    int64 a = stoll(argv[1]);
    int64 b = stoll(argv[2]);
    cout << mult(a,b) << "\n";
    return 0;
}

int64 mult(int64 a,int64 b) {
    if(a == 1) return b;
    if(a & 1) return mult(a >> 1,b << 1) + b;
    return mult(a >> 1,b << 1);
}

```

1.2 Este algoritmo Consiste en dos números A,B:

- Dividir A entre 2, sucesivamente, ignorando el residuo, hasta llegar a la unidad.
- Multiplicar B por 2 tantas veces como veces se ha dividido A entre 2
- Sumar el valor de B cuando el valor de A es impar.

Aquí en el código éste proceso lo observamos en la funcion mult,llamamos recursivamente hasta que A valga 1 y retornamos el valor b, si el numero A es impar se suma el valor de b con el que retorna la llamada anterior sino solamente se vuelve a dividir a A y multiplicamos por 2.



```

lopesc@lopesc-Lenovo-G40-45 ~/Downloads/p3_algoritmos
File Edit View Search Terminal Help
lopesc@lopesc-Lenovo-G40-45 ~/Downloads/p3_algoritmos $ ./sort nums
lopesc@lopesc-Lenovo-G40-45 ~/Downloads/p3_algoritmos $ ./3 nums
at:1641.50
at:1111.50
at:32519.20
at:1288.80
at:24193.40
at:450.00
at:2815.40
th:26215.50    tl:316.70
lopesc@lopesc-Lenovo-G40-45 ~/Downloads/p3_algoritmos $ ./mult 12 40
480
lopesc@lopesc-Lenovo-G40-45 ~/Downloads/p3_algoritmos $

```

```

#include <bits/stdc++.h>
using namespace std;

int main(int arg ,char **c ){
    string n;
    ifstream f(c[1]);
    int M=10,i;
    float rh=.3,bt=0,th=0,tl=0;
    float rl=.1;
    vector <float> at;
    cout<<fixed;

    if(f.is_open()){
        while(getline(f,n)){
            bt=atof(n.c_str());
            if(bt>M){
                at.push_back(bt*(1-rh));
                th=th+bt*rh;
            }
            else{
                at.push_back(bt*(1-rl));
                tl=tl+bt*rl;
            }
            cout<<"th:"<<setprecision(2)<<th<<"\t";
            cout<<"tl:"<<tl<<"\t";
            cout<<"at:"<<at[i]<<"\n";
            i+=1;
        }
    }

    return 0 ;
}

```

1.3 En este último programa donde el objetivo era mejorar y reducir las operaciones en el algoritmo propuesto en el pdf lo que se realizó fue, suponiendo que los valores de los salarios iban a ser leídos desde un archivo, al momento de estar leyendo línea por línea del archivo inmediatamente guardamos en una sola variable el valor de bt, de esta forma dependiendo del valor de bt con respecto a M solamente procedemos a asignar el salario ya calculado con el impuesto en el arreglo At, donde si requerimos almacenar los valores para cada respectivo bt y solo sumamos a th o tl su respectivo porcentaje de impuesto. Las fallas del algoritmo anterior era tener todos los salarios antes y después de los impuestos en un array, recorrer dos veces el array bt y comparar bt con M dos veces para hacer las asignaciones cuando se podía comparar y asignar en un mismo ciclo.