

Tablas en C



29 de Abril de 2021

Esquema de la clase

Veremos la forma de optimizar un código mediante el uso de tablas.

¿Lo pudimos haber usado en el tp de percolación?

```

-----
#include <time.h>

#define S 100
#define I 1000000000
#define E 11
#define SEED 260572
void imprimir(float *tiempos);

int main()
{
    int i,j,aleatorio;
    float duracion_programa,tiempos[S],resultado;
    clock_t c0,c1;
    srand(SEED);
    c0=clock();

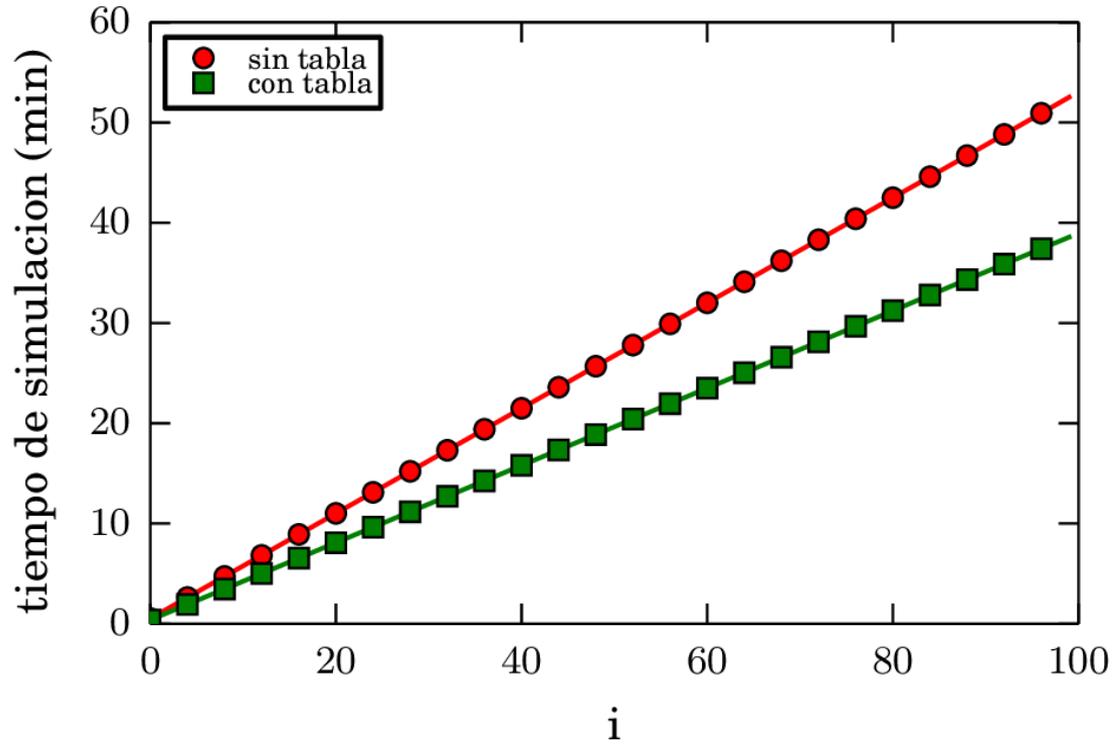
    for(i=0; i<S; i++)
    {
        for(j=0; j<I; j++)
        {
            aleatorio = rand() % E; //número entero entre 0 y 10
            resultado = exp(aleatorio);
        }
        c1=clock();
        duracion_programa = (float)(c1-c0)/CLOCKS_PER_SEC; //en segundos, CLOCKS_PER_SEC=1M
        tiempos[i] = duracion_programa;
    }
    imprimir(tiempos);
    return 1;
}

```

En concreto:

calculamos varias veces la exponencial de un número.

Imprimimos la duración del programa cada I pasos



Siempre conviene usar tablas pero a veces no “ganamos” mucho tiempo.

```
#include <time.h>

#define S 100
#define I 1000000000
#define E 11
#define SEED 260572
void imprimir(float *tiempos);

int main()
{
    int i,j,aleatorio;
    float duracion_programa,tiempos[S],resultado;
    clock_t c0,c1;
    srand(SEED);
    c0=clock();

    for(i=0; i<S; i++)
    {
        for(j=0; j<I; j++)
        {
            aleatorio = rand() % E; //número entero entre 0 y 10
            resultado = exp(aleatorio);
        }
        c1=clock();
        duracion_programa = (float)(c1-c0)/CLOCKS_PER_SEC; //en segundos, CLOCKS_PER_SEC=1M
        tiempos[i] = duracion_programa;
        printf("%i\n",i);
    }
    imprimir(tiempos);
    return 1;
}
```

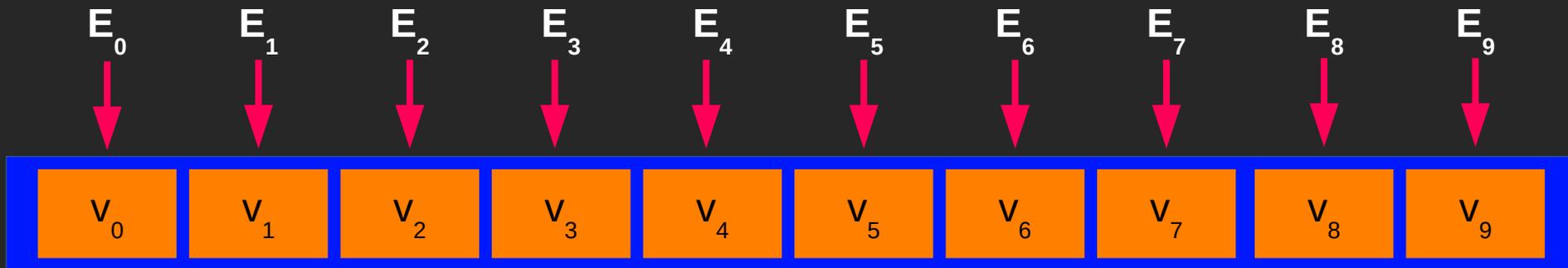
**Estamos calculando
muchas veces los mismos
10 valores.**

**¿Y si los calculamos una
sola vez y los usamos
luego?**

La esencia de las tablas en C

Calculamos todos los posibles valores que puede tomar una variable. Ej. $v_0 = \exp(2 * E_1 + 6)$, $v_1 = \exp(2 * E_2 + 6)$, etc. y los guardamos ordenados en una tabla (puntero).

Si “mido” E y me da E_1 , entonces simplemente fijándome qué hay almacenado en la componente 1, listo (no hago ninguna cuenta).



```

#define S 100
#define I 1000000000
#define E 11
#define SEED 260572
void imprimir(float *tiempos);
void armo_tabla(float *tabla);

int main()
{
    int i,j,aleatorio;
    float duracion_programa,tiempos[S],tabla[E],resultado;
    clock_t c0,c1;
    srand(SEED);
    c0=clock();

    armo_tabla(tabla);

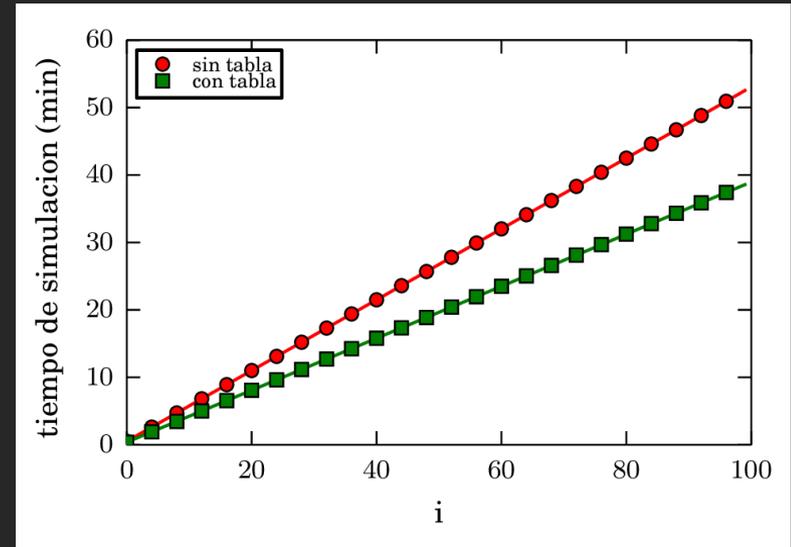
    for(i=0; i<S; i++)
    {
        for(j=0; j<I; j++)
        {
            aleatorio = rand() % E; //número entero entre 0 y 10
            //resultado = exp(aleatorio);
            resultado = tabla[aleatorio];
        }
        c1=clock();
        duracion_programa = (float)(c1-c0)/CLOCKS_PER_SEC; //en seg
        tiempos[i] = duracion_programa;
    }
    imprimir(tiempos);
    return 1;
}

```

```

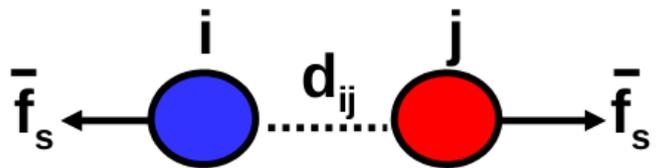
void armo_tabla(float *tabla)
{
    int i;
    for(i=0; i<E; i++) tabla[i] = exp(i);
}

```

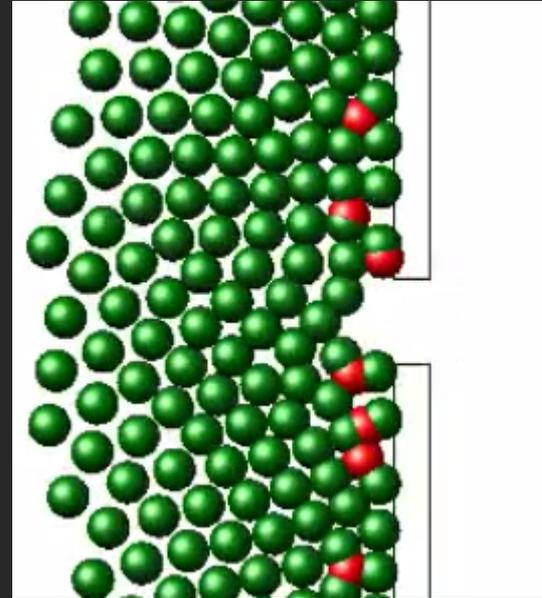


Una aplicación de las tablas

Fuerza social

$$\mathbf{f}_s^{(ij)}(t) = A_i e^{(r_{ij} - d_{ij}(t))/B_i} \hat{\mathbf{n}}_{ij}(t)$$


The diagram shows two particles, labeled 'i' and 'j'. Particle 'i' is a blue circle on the left, and particle 'j' is a red circle on the right. A horizontal dashed line connects their centers, with the label d_{ij} above it. From particle 'i', a black arrow labeled $\bar{\mathbf{f}}_s$ points to the left. From particle 'j', a black arrow labeled $\bar{\mathbf{f}}_s$ points to the right.



High pressures in room evacuation processes and a first approach to the dynamics around unconscious pedestrians, FE Cornes, GA Frank, CO Dorso, Physica A: Statistical Mechanics and its Applications 484, 282-298