

Práctica 3. Divide y vencerás

Ignacio Rodríguez Pérez
ignacio.rodriguezperez@alum.uca.es
Teléfono: +34 697 469 718

22 de diciembre de 2020

1. Describa las estructuras de datos utilizados en cada caso para la representación del terreno de batalla.

Al igual que en la p1, se ha usado una estructura de Celdas (con el valor de la misma), de modo que represente cada una de las casillas del mapa. Dichas celdas, como el enunciado de la práctica así lo pide, se usan montículos y vectores (no se usan listas porque para el acceso en los algoritmos sería más complicado, al no tener métodos de acceso directo).

2. Implemente su propia versión del algoritmo de ordenación por fusión. Muestre a continuación el código fuente relevante.

```
void fusionAux(std::vector<Celda>& v, size_t i, size_t k, size_t j){
    std::vector<Celda> w(j-i+1);
    size_t n = j-i+1, p = i, q = k+1;
    for(int l = 0; l < n; ++l)
        w[l] = ((p <= k) and ((q > j) or (v[p] >= v[q])))? w[l] = v[p++]: v[q++];

    for (int l = 0; l < n; v[i+l] = w[l++]);
}

void sortByFusion(std::vector<Celda>& v, size_t i, size_t j){
    size_t n=j-i+1;
    if(n <= 2){
        if(v[i]<v[j])
            std::swap(v[i],v[j]);
    }else{
        size_t k = i + n/2;
        sortByFusion(v, i, k);
        sortByFusion(v, k+1, j);
        fusionAux(v, i, k, j);
    }
}
```

3. Implemente su propia versión del algoritmo de ordenación rápida. Muestre a continuación el código fuente relevante.

```
size_t pivot(std::vector<Celda>& v, size_t i, size_t j){
    size_t p = i;
    Celda x = v[i];
    for (size_t k = i+1; k <= j; ++k)
        if (v[k] >= x)
            ++p, std::swap(v[k], v[p]);

    v[i] = v[p];
    v[p] = x;
    return p;
}

void quickSort(std::vector<Celda>& v, size_t i, size_t j){
    size_t n = j-i+1;
    if (n<=2){
        if(v[i]<v[j])
            std::swap(v[i], v[j]);
    }else{
        size_t p = pivot(v, i, j);
        if(p>i)quickSort(v, i, p-1);
    }
}
```

```

        if(p<j)quickSort(v, p+1, j);
    }
}

```

4. Realice pruebas de caja negra para asegurar el correcto funcionamiento de los algoritmos de ordenación implementados en los ejercicios anteriores. Detalle a continuación el código relevante.

```

void cajaNegra()
{
    std::vector<Celda> start, copy;

    for(int i = 10; i<=50; i = i+10)
    {
        for(int j = i-1; j >= 0; --j)
            start.push_back(Celda(0,0,j)), copy.push_back(Celda(0,0,j));

        do{ sortByFusion(copy,0,i-1);
        } while (start==copy && std::next_permutation(copy.begin(),copy.end()));
    }
    std::cout << "fusion " << (start==copy ? "si":"no")<< " funciona." << std::endl;

    start.clear(),copy.clear();

    for(int i = 10; i<=50; i = i+10)
    {
        for(int j = i-1; j >= 0; --j)
            start.push_back(Celda(0,0,j)), copy.push_back(Celda(0,0,j));

        do{ quickSort(copy,0,i-1);
        } while (start==copy && std::next_permutation(copy.begin(),copy.end()));
    }
    std::cout << "ord rapida " << (start==copy ? "si":"no")<< " funciona." << std::endl;
}

```

5. Analice de forma teórica la complejidad de las diferentes versiones del algoritmo de colocación de defensas en función de la estructura de representación del terreno de batalla elegida. Comente a continuación los resultados. Suponga un terreno de batalla cuadrado en todos los casos.

El algoritmo sin preordenación es de orden $O(n^2)$, mientras que, los de ordenación, son más eficientes, ya que su orden es $O(n \cdot \log(n))$; no obstante, la grafica muestra lo contrario.

6. Incluya a continuación una gráfica con los resultados obtenidos. Utilice un esquema indirecto de medida (considere un error absoluto de valor 0.01 y un error relativo de valor 0.001). Es recomendable que diseñe y utilice su propio código para la medición de tiempos en lugar de usar la opción *-time-placeDefenses3* del simulador. Considere en su análisis los planetas con códigos 1500, 2500, 3500,..., 10500, al menos. Puede incluir en su análisis otros planetas que considere oportunos para justificar los resultados. Muestre a continuación el código relevante utilizado para la toma de tiempos y la realización de la gráfica.

Todo el material incluido en esta memoria y en los ficheros asociados es de mi autoría o ha sido facilitado por los profesores de la asignatura. Haciendo entrega de este documento confirmo que he leído la normativa de la asignatura, incluido el punto que respecta al uso de material no original.

