

Algoritmos y Estructuras de datos

Ejercicios previos al segundo examen parcial

Problemas de ordenamiento

- Escribe una función que tome dos enteros y pueda ser usada por `std::sort` como predicado para ordenar enteros de la siguiente forma: los enteros negativos deben quedar primero y los no negativos después, desempataando enteros del mismo grupo por valor absoluto de menor a mayor. Por ejemplo, usar esta función con `std::sort` sobre la secuencia `{ 8, -7, 1, -2, 6, -4, 0 }` debe resultar en `{ -2, -4, -7, 0, 1, 6, 8 }`. Puedes usar cualquier utilidad de la biblioteca de C++.

```
bool signo_magnitud(int a, int b) {
    if ((a >= 0) != (b >= 0)) {
        return a < b;
    } else {
        return abs(a) < abs(b);
    }
}
```

- Escribe una función que tome dos alumnos y pueda ser usada por `std::sort` como predicado para ordenar alumnos por calificación de mayor a menor; si hay alumnos que tienen la misma calificación, éstos deben ordenarse por número de lista de menor a mayor.

```
struct alumno {
    int lista, calif;
};

bool predicado(alumno a, alumno b) {
    if (a.calif != b.calif) {
        return a.calif > b.calif;
    } else {
        return a.lista < b.lista;
    }
}
```

- Algunos de los problemas en omegaUp están resueltos en las notas.