Zajęcia C-1: “Sortowanie za pomocą STL-a”

Cel zajęć i efekty uczenia

Główne cele zajęć / materiał do opanowania:

- Używanie funkcji sort(), pojęcie funkcji porównującej (komparatora)
- Pojęcie klasy/struktury/rekordu

Dodatkowe cele:

- Operacje na liczbach rzeczywistych
- Wprowadzenie do algorytmów geometrycznych: sortowanie kątowe w najprostszej wersji

Zadania do rozwiązania na sprawdzarce

Panorama
Dane są punkty na płaszczyźnie (szczyty górskie). Jaka będzie ich kolejność widziana z punktu (0,0)?

Plan zajęć

Szacunkowy czas trwania: 2 godziny lekcyjne.

1. Funkcja sort() w najprostszej postaci - na tablicy liczb całkowitych
    - *Składnia funkcji na tablicy i na STL-owym wektorze*
    - *Łagodne wprowadzenie do iteratorów: sugeruję podejście “zmiennej pokazującej pozycję w tablicy”, bez wchodzenia w szczegóły - trzeba jednak zatrzymać się przy “pozycji za ostatnim elementem”*
2. Struktura w C++
    - *Omówienie na przykładzie punktów w zadaniu (trzy pola: odcięta, rzędna, numer)*
    - *Na razie tylko pola - opcjonalnie można opowiedzieć o klasach i metodach, chociaż w tym momencie potrzbne nie będą*
3. Sortowanie z własnym komparatorem
    - *STL-owa składnia: sort(początek, koniec, funkcja_porównująca)*
    - *Przykład: sortowanie według pierwszej współrzędnej/leksykograficzne*
    - *Własności, jakie musi spełniać komparator (nie może być a < a, nie może być cyklu, etc.)*
4. Sortowanie kątowe - wzory geometryczne
    - *Różnica między sortowaniem leksykograficznym (“od lewej do prawej”) a kątowym - konieczne może być dłuższe zatrzymanie się w tym miejscu, jako że typowo kolejność kątowa nie jest intuicyjna.*
    - *Po zrozumieniu przez uczniów idei kąta nachylenia, należy pokazać (wyprowadzić) wzór geometryczny: punkt A jest na lewo od B, jeśli A.x/A.y < B.x/B.y.*

5. Ostateczna postać funkcji-komparatora
    - *Nie wchodząc w szczegóły, warto zauważyć, że unikamy przy programowaniu liczb rzeczywistych, a szczególnie porównania: zamiast A.x/A.y < B.x/B.y dużo lepiej napisać A.x * B.y < A.y * B.x.*
    - *Kwestia punktów leżących w jednej linii - przy równym kącie najpierw jest leżący bliżej*
    - *Opcjonalnie można zastanowić się, jakie pojawiłyby się problemy, gdyby punkty leżały na całej płaszczyźnie, a nie tylko w górnej połówce.*
    - *Kwestia techniczna: uważać na duże liczby! Niezmieszczenie się w 32-bitowym typie **int** przy porównywaniu kątowym to jeden z najpopularniejszych błędów popełnianych na zawodach informatycznych.*