

Zajęcia C-4: "Kolejka priorytetowa"

Cel zajęć i efekty uczenia

Główne cele zajęć / materiał do opanowania:

- Struktura kolejki priorytetowej
- [opcjonalnie] Kopiec i operacje na kopcu, sortowanie przez kopcowanie

Dodatkowe cele:

- Przeładowanie operatorów (na przykładzie operatora "<") w C++

Zadania do rozwiązania na sprawdzarce

Wybory

Jest n partii, każda otrzymała pewną liczbę głosów w wyborach. Rozdzielić m mandatów [metodą d'Hondta](#).

Plan zajęć

Szacunkowy czas trwania: 2 godziny lekcyjne.

1. Kolejka priorytetowa

- Na dobrą sprawę kolejka priorytetowa nie różni się za wiele od zwykłej kolejki, przez co nie trzeba zbyt wiele tłumaczyć. Również STL-owa kolejka `priority_queue` działa bardzo podobnie do `queue`.
- Kolejka na własnych strukturach: przeładowanie operatora "<"
- Można (teoretycznie) robić kolejkę z własnym komparatorem (funkcją porównującą), ale w C++ ma to wyjątkowo dziwną składnię (inną niż przy sortowaniu) i jest należy skomplikowana, że raczej należy ją na tym etapie odradzać.
- Złożoność kolejki: $\log n$ na operację, gdzie n to liczba elementów, które aktualnie są na kolejce.

2. Zadanie "Wybory"

- Nie możemy stworzyć całej tabelki, bo zużyłoby to zbyt wiele czasu i pamięci komputera. Algorytm powinien działać w czasie $O(n \log n + m \log n)$, gdzie n to liczba partii, m - mandatów.
- Intuicja do metody d'Hondta: wszystkie partie liczą swoje głosy. Ta, która ma najwięcej, otrzymuje mandat, po czym liczba jej głosów zmniejsza się do połowy oryginalnej. Kolejna partia z największą liczbą głosów otrzymuje mandat, i tak dalej. Jeśli partia z oryginalną liczbą głosów x dostała mandat numer k , to liczba jej głosów będzie teraz wynosiła $x/(k+1)$.
- Ta intuicja wprost przekłada się na algorytm: wszystkie partie są na kolejce priorytetowej, wyciągamy z niej partię o największej liczbie głosów, po czym z powrotem dorzucamy do kolejki z "nową" liczbą głosów.

- *Treść zadania narzuca dość skomplikowany operator porównania, warto go dokładnie omówić z uczniami.*
- 3. [opcjonalnie] Sortowanie i kopiec
 - *Warto zauważyć, że używając kolejki priorytetowej da się napisać sortowanie, które będzie działało w $O(n \log n)$.*
 - *Ambitnym uczniom można w tym momencie objaśnić strukturę kopca i operacje kopcowania - ja typowo nie wyjaśniam tego uczniom, jako że bardzo rzadko (nawet na Olimpiadzie Informatycznej) zachodzi potrzeba rozumienia struktury kopca, albo pisania go samodzielnie. Podobne struktury będą wprowadzane przy okazji drzew przedziałowych.*