Zajęcia A-1: “Co to jest algorytmika?”

Cel zajęć i efekty uczenia

Główne cele zajęć / materiał do opanowania:

- Pojęcie algorytmu i programu, interakcja między człowiekiem a maszyną, konieczność przestrzegania formalnej poprawności programu
- Intuicyjne podejście do złożoności obliczeniowej, algorytmy wolne i szybkie

Dodatkowe cele:

- łagodne wprowadzenie do wyszukiwania binarnego
- pojęcie logarytmu dwójkowego

Plan zajęć

Szacunkowy czas trwania: 2 godziny lekcyjne (orientacyjnie, patrz “Uwaga” poniżej).

Uwaga: Zajęcia wprowadzające w algorytmikę i programowanie (A-1, A-2) muszą bardzo mocno zależeć od tego, jak liczna jest grupa, jaki ma początkowy poziom zaawansowania, czy zajęcia odbywają w pracowniach komputerowych, ile jest komputerów i jakie mają oprogramowanie. Ogromne znaczenie - większe niż przy następnych zajęciach - mają indywidualne preferencje nauczyciela i własny sposób “dotarcia” do uczniów. Dlatego też w wypadku tych zajęć szacunkowy czas ich trwania oraz podany plan lekcji należy traktować, jeszcze bardziej niż przy dalszych zajęciach - jako punkt wyjścia do rozwinięcia własnych metod.

1. Dyskusja: co to jest “algorytm”?
    - *Dyskusja ma na celu pobudzenie uczniów do samodzielnego myślenia, nie wypracowanie formalnej “definicji algorytmu”. Tak naprawdę, nie istnieje jedna, obowiązująca “definicja algorytmu”, ani nawet żadna definicja, która byłaby jednocześnie zrozumiała i znacząca. Najlepsze przybliżenie, jakie mogę zalecić to “ciąg poleceń sformułowany w języku zrozumiałym dla maszyny”.*
2. Jak zbudowany jest komputer i jak wykonuje nasze polecenia? Jak formułować polecenia tak, żeby były zrozumiałe dla maszyn? Co to jest “język programowania”?
    - *Ważne jest podkreślenie, że komputer nie “domyśla się” i nie potrafi zrozumieć “o co nam chodziło”. Musi dostawać proste polecenia, odwołujące się tylko do pojęć z wąskiej listy “zrozumiałych dla niego”.*
    - *Można omówić, ale bez wchodzenia w szczegóły i definicje, pojęcia:*
        - *pamięci operacyjnej - podkreślając, że jest podzielona na komórki, i de facto przechowuje wyłącznie liczby*
        - *wejścia i wyjścia programu - najlepiej na przykładach prostych algorytmów (np. dodających dwie liczby)*
        - *programu - ciągu instrukcji umieszczonych w pamięci, które wykonują się zawsze w podanej kolejności (kolejnością można manipulować za*

pomocą instrukcji, ale nigdy nie jest przypadkowa, zawsze wykonuje się też jedna instrukcja naraz)

3. Gra w “za dużo, za mało” i wyszukiwanie binarne jako przykład algorytmu
    - *Dawna radiowa gra w “za dużo, za mało” polegała na zgadnięciu pewnej (tajnej) kwoty pieniędzy przez słuchaczy dzwoniących do radia. Słuchacz podawał kwotę, a system odpowiadał, czy kwota jest za duża, czy za mała. Trafienie w dokładną kwotę oznaczało wygranie jej przez dzwoniącego.*
    - *Zakładamy, że jesteśmy jedynym dzwoniącym, a kwota jest między 1 a 1000 złotych. Jak wygrać ją za pomocą co najwyżej 1000 telefonów? Jak wygrać ją w mniejszej liczbie (10) telefonów?*
4. Złożoność algorytmu - łagodne wprowadzenie
    - *Dlaczego algorytm “dzielący na pół” jest lepszy? Omówienie (nieformalne), czym jest złożoność pesymistyczna - zakładamy, ze system zachowuje się zgodnie z regułami (kwota naprawdę jest między 1 a 1000), ale nasz algorytm zawsze ma pecha - kwota jest taka, żeby telefonów było jak najwięcej.*
5. Logarytm dwójkowy
    - *Ile będzie potrzeba telefonów, jeśli kwota jest między 1 a n? Pojęcie logarytmu dwójkowego.*
    - *Nie ma potrzeby ogólnego definiowania logarytmu o dowolnej podstawie, ani nawet żadnej formalnej definicji logarytmu - przez większość kursu wystarczy nam logarytm dwójkowy zaokrąglony do liczby całkowitej. Oczywiście należy nadmienić, że to uproszczone pojęcie i “kiedyś” wprowadzimy je w całości.*