* Sortowanie przez scalanie: z ćwiczeniem praktycznym na patyczkach
* Lekcja druga: wprowadzenie do sortowania szybkiego
* Lekcja trzecia: implementacja sortowania szybkiego

Uwagi:

* Przedstawienie idei algorytmu sortowania szybkiego jest względnie proste: mniejsze na lewo, większe na prawo, powtarzaj. Diabeł tkwi w szczegółach.
* Ideę algorytmu można zaprezentować na przykładzie sortowania książek pod względem wysokości: wybieramy środkową, niższe na lewo, wyższe na prawo, powtarzamy.
* Pytanie 1: w jaki sposób przedstawić „istotność” tego algorytmu? Może porównać z innymi algorytmami sortowania.
* Pytanie 2: co jest najważniejsze w algorytmie sortowania szybkiego? Co uczniowie powinni wynieść z lekcji?
* Pytanie 3: jaki jest cel lekcji? Czy chcemy, aby wszyscy potrafili zaimplementować algorytm w wybranym języku? Czy może lepiej przenieść implementację na coś innego, bardziej „praktycznego”?
* Spostrzeżenie 1: być może implementacja algorytmu quicksort stanowi pewną barierę – nie wszyscy będą w stanie to zrozumieć.
* Spostrzeżenie 2: zalety algorytmu quicksort są niewidoczne na małych danych – dużo wysiłku, marny rezultat. Z „zewnątrz” sortowanie przez wybieranie wydaje się być szybsze i prostsze.
* Uwaga 1: być może należałoby pokazać, dlaczego algorytm quicksort jest tak wyjątkowy i istotny (działa w miejscu i co jeszcze)?
* Uwaga 2: dla ucznia algorytm quicksort w porównaniu do sortowania przez wybieranie może wydawać się nienaturalny i być niepotrzebnym utrudnieniem. W jaki sposób sprawić, by był bardziej „naturalny”? Czy da się zrobić coś podobnego jak w przypadku rekurencji i przeszukiwania pudełek? Czy istnieje przykład zastosowania/problemu, w którym quicksort jest „naturalnym” rozwiązaniem?
* Pytanie 4: gdzie, poza komputerem, stosujemy sortowanie? Biblioteka? Możemy sortować książki, karty (w trakcie gry). Co jeszcze?
* Czy istnieje sytuacja, którą można zaprezentować poprzez prostą analogię, w której algorytm sortowania szybkiego jest znacznie lepszym pomysłem niż sortowanie przez wybieranie? Dobrze, żeby chodziło o coś innego niż sama złożoność. Ludzie mają tendencję do twierdzenia, że współczesne komputery są w stanie wszystko policzyć w krótkim czasie.
* Pytanie 5: jak, w sposób prosty, przedstawić nie tylko ideę sortowania szybkiego, ale także samego przesuwania elementów dokonywanego w miejscu? Może jako pivot określić element brzegowy? Może przykład z książkami: zaczynamy od lewej (pierwsza książka służy za pivot i określa próg wysokości) i idziemy w prawo. Jak napotkamy wyższą od pivota, to jest ok (książka jest na swoim miejscu – po prawej stronie), idziemy dalej w prawo. Jeżeli napotkamy niższą od pivota, to zamieniamy miejscami – tę niższą i pierwszą na prawo od pivota od niego wyższą. Jest to o tyle łatwiejsze, że przesuwamy się tylko w jedną stronę – w prawo. Możemy wykorzystać dwa palce (wskaźniki) do wskazywania, na które książki patrzymy. Lewa dłoń dotyka tylko pivota i od niego niższych. Prawa dłoń przeskakuje po wyższych, ale zatrzymuje się na niższych wyznaczając elementy do zamiany. Na koniec lewa dłoń wyznacza miejsce podziału tablicy. Można zacząć od przykładu, w którym drugim elementem (zaraz obok pivota) jest książka wyższa od pivota (tak będzie łatwiej).

Pomysł inny: wyszukiwanie binarne na przykładzie książki telefonicznej (albo innego, uporządkowanego spisu, teraz właściwie nie korzysta się z książek telefonicznych).